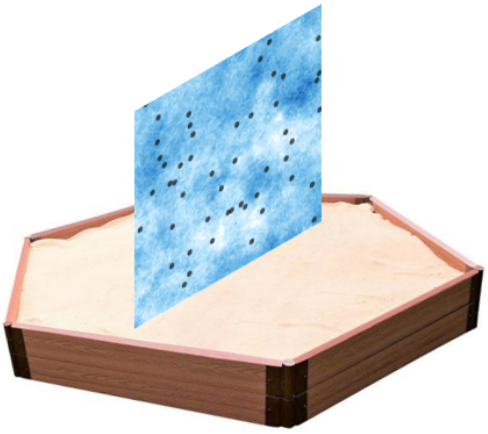
The spatial prediction sandbox

Investigating the use of spatially-explicit modelling and cross-validation strategies in spatial interpolation machine learning problems



Carles Milà

Institute for Geoinformatics, University of Münster

Supervisor: Hanna Meyer (WWU)

Co-supervisors: Edzer Pebesma (WWU) and Jorge Mateu (UJI)

A thesis submitted for the degree of

Master in Geospatial Technologies

2021

Acknowledgements

I would like to express my gratitude to the supervisors of my Master Thesis for their dedication, ideas, feedback, and support. I would like to especially thank Prof. Dr. Meyer for her continuous guidance from the very inception of the project, helping me to define the scope of the research, find relevant literature, make tough methodological decisions, narrow down the uncountable possible paths to explore, and welcoming me to her research group.

I am very grateful to the GEOTECH Master coordinators and especially to Prof. Dr. Painho and Dr. Brox for organising the beautiful program I enjoyed in Lisbon and Münster. I also would like to acknowledge the financial support of Erasmus Mundus, without which I would not have been able to be a part of this adventure.

Finally, I would like to give my warmest thanks to my parents, who once again welcomed me and helped me stay safe in a year none of us will forget; to my new Geomundus friends who have been a continuous source of inspiration, laughter, and energy; to my sister and friends, who have supported me when I needed it most; and especially to Ferran, who put up with my maps and statistics and stayed at my side despite being always on the move.

Abstract

Machine Learning (ML) methods are increasingly used for spatial interpolation and different strategies have been proposed to introduce space into the modelling and validation phases. Nevertheless, a comparison of these methods under different landscape autocorrelation ranges and sampling designs is still missing. This Master Thesis investigates under which scenarios spatially-explicit ML modelling and validation strategies are appropriate for spatial interpolation problems.

We designed a framework that allowed us to simulate predictor and outcome spatial fields with different autocorrelation ranges, as well as samples with different number of points and distributions. With these data, we tested different non-spatial and spatially-explicit (coordinates, EDF, RFsp) Random Forest ML models and evaluated them using the simulated surfaces as well as different standard (Leave-One-Out, LOO) and spatially-explicit (spatial buffer LOO, sbLOO) Cross-Validation (CV) strategies. We developed a new method called Nearest Distance Matching (NDM) to estimate the appropriate radius for sbLOO CV for spatial interpolation based on sample distribution and landscape range, and compared it to state-of-the art methods for radius search, only based on range.

While for short ranges non-spatial models were superior to spatially-explicit models regardless of the sample size and distribution; for long ranges, spatial models performed better under regular and random sampling designs, but not clustered and non-uniform. CV results indicated that although LOO correctly estimated model performance under random designs, it yielded overestimated errors for regular samples and underestimated errors for clustered and non-uniform designs under long ranges. Results of sbLOO combined with NDM correctly addressed error underestimation of LOO in clustered and non-uniform samples, whereas sbLOO based solely on the range resulted in error overestimation for all designs under long ranges.

This Master Thesis provides important insights to the field of predictive mapping: it elucidates in which cases spatially-explicit methods may be preferred, and establishes that state-of-the-art approaches for spatial CV designed to assess model transferability are not suited for spatial interpolation and proposes an alternative.

Abbreviations

CSR: Complete Spatial Randomness

CV: Cross-Validation

EDF: Euclidean Distance Fields

LOO: Leave-One-Out

MAE: Mean Absolute Error

ML: Machine Learning

NDM: Nearest Distance Matching

RF: Random Forest

RFsp: Random Forest for spatial prediction

RMSE: Root Mean Square Error

sbLOO: spatial buffer Leave-One-Out

SD: Standard Deviation

SSE: Sum of Squares Error

Contents

Chapter 1

Introduction

1.1 Spatial interpolation overview

The availability of spatially continuous data of environmental variables, as well as any other variables varying continuously in space, is critical in a broad range of research fields and practical applications. To name only a few examples, continuous maps of meteorological variables such as air temperature and precipitation (Fick and Hijmans, 2017) are needed to model species distribution and biodiversity (Veloz, 2009); soil maps (Hengl et al., 2017) are important to model phenomena such as food productivity (Folberth et al., 2016); while air pollution maps (Beelen et al., 2013) are required for exposure assessment in environmental epidemiology (Raaschou-Nielsen et al., 2013) as well as policy-making.

Spatially continuous fields mostly use a raster model consisting on an array of cells forming a grid, in which the value of each pixel represents the value of the variable at a given location (Longley et al., 2005). Since the sampling process is often complex and costly (e.g. meteorological stations, soil profiles), only a limited set of geolocated measurements are typically available. In this context, spatial prediction is the process whereby the value of a variable measured at a limited set of locations is predicted for an unmeasured location (Longley et al., 2005). In this Master thesis, we will focus in spatial interpolation in the geographic sense, defined as prediction at unsampled locations within the study area from which the samples are drawn (otherwise it is called geographic extrapolation) (Li and Heap, 2014)[1]. By predicting at unmeasured locations, spatial interpolation allows to construct continuous surfaces of variables discretely and finitely sampled (Li and Heap, 2014).

The process of performing a spatial interpolation can generally be divided into different phases (Figure 1.1):

1. Sample design and collection: When possible, sampling campaigns are designed and the target variable (i.e. the outcome) is measured. In many cases, sampling locations are defined *a priori* (e.g. fixed monitoring stations, already existing samples).
2. Covariate collection: Spatially continuous predictors of the outcome are collected. Those typically consist of remotely-sensed data (e.g. optical satellite bands and spectral indices,

[1]Note that there is an alternative definition for interpolation and extrapolation referring to the predictor space (Meyer and Pebesma, 2020; Roberts et al., 2017)

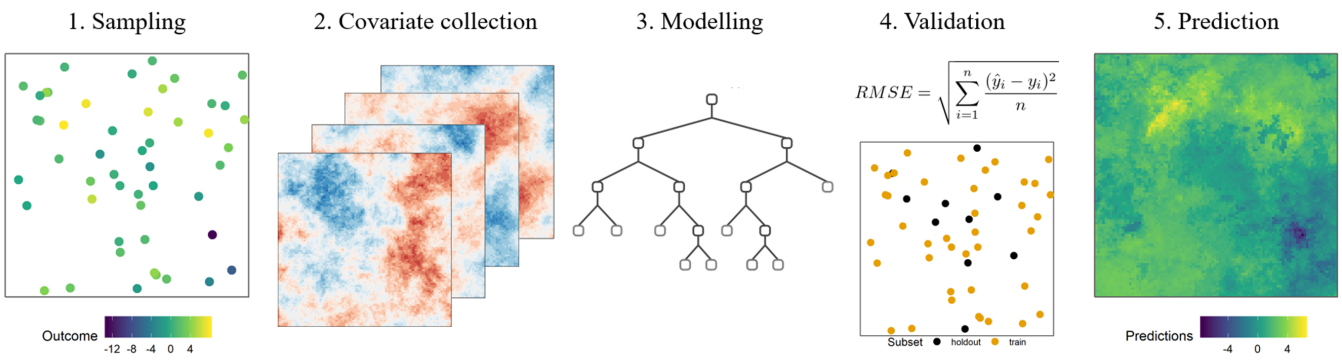


Figure 1.1: Spatial interpolation workflow overview.

Phiri and Morgenroth 2017), terrain variables (e.g. digital elevation models, slope, aspect; Meyer et al. 2016), Geographic Information Systems (GIS) derived variables (e.g. distance to major road, Beelen et al. 2013), or results of previous modelling endeavours (e.g. deterministic modelling and land cover classifications, de Hoogh et al. 2018). Layers are stacked and the information at the sampling points locations is extracted.

3. Modelling: After an exploratory analysis, data are modelled using geostatistical and non-geostatistical spatial interpolation methods (Li and Heap, 2014), which recently include Machine Learning (ML) and hybrid approaches (Li et al., 2011).
4. Model validation: The performance of the model is assessed using a test dataset or resampling techniques in order to estimate the accuracy of the predicted map (de Hoogh et al., 2018). Further residual validation (model hypotheses, residual spatial autocorrelation) may be carried out (Hengl et al., 2018).
5. Map creation: If the results of the validation are satisfactory, the model can be used to predict the outcome at all locations of the study area and thus create a continuous surface. For some modelling methods, additional surfaces estimating the variability of the predictions can also be created (Hengl et al., 2018).

1.2 Spatial interpolation methods

Methods for spatial interpolation have been classified into geostatistical, non-geostatistical, and combined groups (Li and Heap, 2014). Geostatistics combines statistical modelling of large-scale variation (i.e. the mean structure) and small-scale variation (i.e. the error structure) under a series of assumptions (Cressie, 2015). Non-geostatistical approaches have traditionally included deterministic (e.g. Nearest Neighbours, Triangulation, Inverse Distance Weighted) and stochastic (e.g. linear regression, spline interpolation) methods (Li and Heap, 2014). Combined approaches couple methods from the two other groups, such as the well-known Regression Kriging method (Hengl et al., 2007).

Even though these methods have a long tradition and are ubiquitous in the literature (Hengl et al., 2009), more recently, ML models have been increasingly used in the geosciences due to their ability to capture complex relationships and to handle highly multivariate problems (Lary et al., 2016). Although the list of available ML models is extensive (Hastie et al., 2009), the most used ML models for spatial prediction are classification and regression trees, Random Forest (RF), support vector machines, and neural networks (Wylie et al., 2019).

One limitation of standard ML models for spatial prediction is that they implicitly assume observations to be independent and thus ignore the spatial structure of the data (Xie et al., 2017). The first attempts to take space into consideration combined ML with other approaches, e.g. by applying Inverse Distance Weighting or Ordinary Kriging to the residuals of the ML model (Li et al., 2011). In the past few years, a series of extensions of purely-based ML models for spatial prediction have been recently proposed, ranging from simply adding coordinate fields (x, y) as predictors (e.g. Cracknell and Reading 2014), to more complex proposals adding distance fields or spatial lags as features in the ML models (Behrens et al., 2018; Hengl et al., 2018; Li et al., 2017; Sekulić et al., 2020). Although all of the methods suggested use different approaches, they all ultimately aim to incorporate space into the model in a single step so that (residual) spatial dependencies can be captured and therefore the performance of the models can be improved.

A second problem that has been identified in the use of ML for predictive mapping is spatial overfitting. Briefly, the inclusion of highly autocorrelated predictors not causally related to the process being modelled (e.g. coordinate fields) may result in ML models that are able reproduce the training data but fail at predicting new observations (Meyer et al., 2019). Therefore, a trade-off between spatially-explicit ML approaches using highly autocorrelated geographic predictors and spatial overfitting exists.

1.3 Spatial cross-validation methods

As indicated in section 1.1, a key step in the interpolation problem is model validation, during which the error/accuracy of the interpolated map generated by the model is estimated using separate test data or resampling techniques. Standard Cross-Validation (CV) methods (e.g. k-fold, Leave One Out (LOO), and repeated holdout CV) applied to spatial prediction have been acknowledged to produce underestimation of the error in many applications (e.g. Ploton et al. 2020). The main problem of random CV techniques is that they assume train and hold-out data to be independent although this is hardly the case in spatially structured environments (Roberts et al., 2017). Hence, using standard techniques to evaluate model transferability, i.e. to assess the error associated with using a model in an area different than where it was trained, might lead to overoptimistic results (Telford and Birks, 2009; Wenger and Olden, 2012).

In order to overcome this issue, a series of CV techniques have been proposed for spatial (Valavi et al., 2019) and spatio-temporal (Meyer et al., 2018) data. While these dedicated techniques can make a strong impact in the performance estimation of the predicted maps (Meyer et al., 2019), they do not seem to play a major role in the hyperparameter search during the ML model fit (Schratz et al., 2019). A key aspect of these methods is how to create the non-random groupings used in the CV process, which depend on one or several parameters and can strongly impact results (Roberts et al., 2017).

1.4 The role of sampling design and spatial autocorrelation

Even though sampling designs play a significant role in spatial prediction problems, the distribution of the samples has been seldom explored in applications. Having more dispersed samples often results in a better model performance than clustered designs due to the increased information available in the samples (Cracknell and Reading, 2014). Indeed, systematic (Rocha et al., 2020) and feature-based (Wadoux et al., 2019) sampling have been recommended to obtain optimal spatial prediction results in ML-based applications. Moreover, it has been acknowledged that the underestimation of the error when using standard CV methods may be more severe under clustered designs (Meyer et al., 2019). Finally, using spurious variables with high spatial autocorrelation as predictors under clustered designs has been suggested to potentially trigger overfitting and thus lead to suboptimal models (Hengl et al., 2018; Meyer et al., 2019).

The degree of spatial autocorrelation of outcome and predictors may also affect the modelling and validation strategies in ML-based spatial prediction, yet it has been rarely analysed in applied studies. Explicit spatial modelling may lead to better prediction results than standard ML models in situations where (residual) spatial dependence is large (Rocha et al., 2019). Furthermore, higher dependence levels might lead to an increased degree of error underestimation when using standard CV techniques (Rocha et al., 2018). In fact, parameters of spatial CV methods have been suggested to be based on outcome and/or residual spatial autocorrelation to ensure independence between training and hold-out data (Roberts et al., 2017).

1.5 Knowledge gap, aim, and research questions

In this introduction we have seen how ML models are increasingly used for spatial interpolation and how different methods have been proposed to introduce space directly into the ML models by using different sets of predictors. However, a comparison of their performance under different factors that may impact them, namely spatial autocorrelation and sampling design, is still missing. Available simulation studies looking at similar issues did not examine any of the recently proposed spatially-explicit ML models (Liao et al., 2018; Rocha et al., 2019; Rocha

et al., 2020), or did not take into account the influence of sampling or spatial autocorrelation (Hengl et al., 2018; Sekulić et al., 2020).

Likewise, there is hardly any evidence on the underestimation of the error when using standard CV techniques for spatial interpolation under different sampling designs (Rocha et al., 2020). Furthermore, current recommendations on the parameter choice for spatial CV methods are solely based on autocorrelation range with the objective of obtaining independent training and held-out data (Valavi et al., 2019; Wenger and Olden, 2012). While this may be appropriate for model transferability, we think it may not be adequate for interpolation in the geographic space, where the prediction locations may not be independent of the training data and therefore independence between held-out and train observations during CV is neither needed nor desired.

The aim of this master Thesis is to offer guidance to spatial prediction practitioners on which cases the recently proposed spatially-explicit ML models and validation strategies might be appropriate for spatial interpolation problems, so that maps developed with these techniques are as accurate as possible and have reliable estimates of their performance.

In order to do so, we want to develop a simulation study so that different models and validation strategies for spatial interpolation can be evaluated under several spatial autocorrelation and sampling scenarios (Figure 1.2) in order to answer the following research questions:

1. Under which conditions are spatially-explicit ML models appropriate?
2. Under which conditions are spatially-explicit CV methods appropriate?
3. Under which conditions are state-of-the-art CV methods designed for spatial extrapolation also suited for interpolation, and is there a CV strategy that can be used across sampling designs and degrees of spatial autocorrelation?

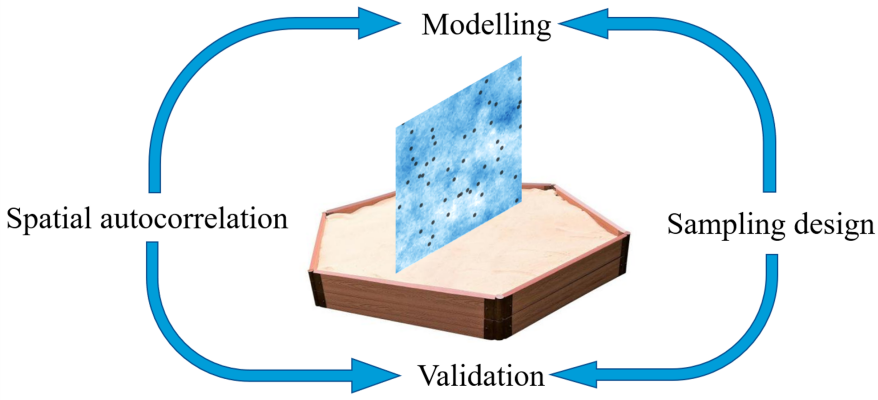


Figure 1.2: The spatial prediction sandbox research questions.

Chapter 2

Literature review

2.1 Machine learning methods for spatial interpolation

2.1.1 Non-spatial models

As introduced in section 1.2, standard non-spatial ML models have been frequently used in the spatial prediction literature even though they implicitly assume data to be independent and ignore their spatial structure (Xie et al., 2017). From all the ML methods that have been developed, we will turn our attention to the one model that has been mostly used in the methodological research on ML-based spatial prediction (e.g. Behrens et al. 2018; Georganos et al. 2019; Hengl et al. 2018; Meyer et al. 2019; Sekulić et al. 2020) and that is one of the mostly used in applications as well (Wylie et al., 2019): The Random Forest (RF) model.

2.1.1.1 The Random Forest (RF) model

The RF algorithm was introduced by Breiman (2001) and is plainly explained in Hastie et al. (2009). Briefly, RF builds on the idea of bagging, in which many (classification/regression) models are fitted on bootstrap samples of the original dataset, and then results are averaged in order to reduce the variance. A usual choice for the model is that of a classification or regression tree, which generally have large variance and low bias. RF builds on that idea by further reducing the variance by decorrelating the trees, which is achieved by taking a random selection of the features in each tree (Figure 2.1). The three hyperparameters of the model are: the number of trees to grow, the minimum node size, and most importantly the fraction of random predictors to select for each tree, which can be all tuned using a grid search.

Since RF is a tree-based ensemble method, it can handle non-linearities, interactions, and feature selection to a certain degree; but deals with extrapolation poorly in regression problems. RF is known to be fairly robust to overfitting in the covariates. Finally, an interesting feature of RF is that variable importance statistics can be calculated based on the Out-Of-Bag samples, i.e. the data records that have been not selected in a particular bootstrap sample.

2.1.1.2 Ignoring residual spatial autocorrelation

Standard ML methods, including RF, may be suboptimal as they do not capture residual spatial dependencies that have not been accounted for in the covariates (Hengl et al., 2018). As a proof

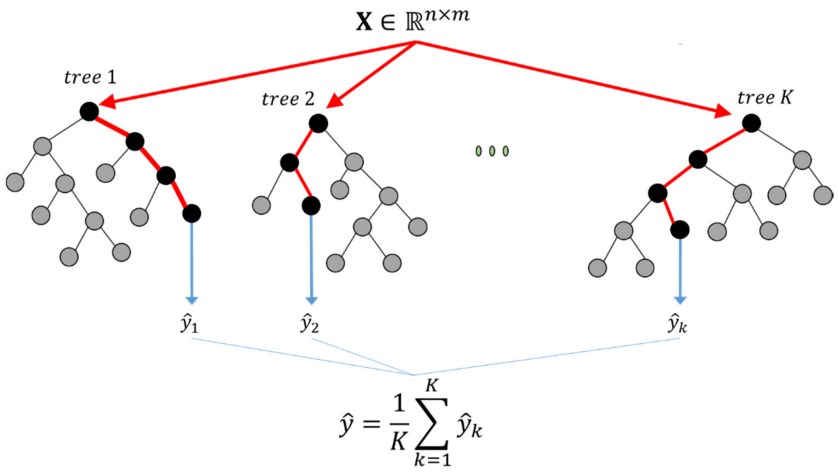


Figure 2.1: RF for regression problems where n is the number of observations, m is the number of features, k is the number of trees. Source: Aldrich (2020).

of that, the study by Rocha et al. (2019) showed that, for spatial interpolation problems with strong spatial dependencies, a simple regression model dealing with spatial autocorrelation via stochastic partial differential equations provided better results than complex non-spatial ML models with a much larger number of covariates.

In the field of geostatistics, residual dependencies are generally dealt with using the regression Kriging method, where a trend (mean structure) is fitted on the response using auxiliary predictors in a regression model, and then simple kriging is used on the residuals (error structure) (Hengl et al., 2007). But what about ML methods? A possible solution is to take a hybrid approach such as those included in the study by Li et al. (2011), where large-scale variation ML modelling is combined with small-scale geostatistical (e.g. Ordinary Kriging) or deterministic (e.g. Inverse Distance Weighting) interpolation on the residuals. However, there is interest in the development of purely ML-based spatially-explicit methods that are able to deal with the full predictive process in a single step.

2.1.2 Spatially-explicit models

Spatially-explicit ML learning models aim to describe covariates and residual spatial autocorrelation jointly in a single step: the ML model (Hengl et al., 2018). With that purpose, a considerable number of ML model extensions for predictive mapping have been proposed in the literature (Behrens et al., 2018; Hengl et al., 2018; Li et al., 2017; Sekulić et al., 2020). Amongst those, we will focus on the three of them that have been the most used and/or have had the largest impact in the spatial prediction literature (measured in the number of citations): adding coordinate fields as predictors, Euclidean Distance Fields (EDF), and Random Forest for spatial prediction (RFsp).

2.1.2.1 Coordinate fields

The eldest, most used, and simplest option is to add geographic coordinate fields (x,y) in the predictor space as two additional covariates in the ML model (Figure 2.2). Even though many studies have used this approach in very distinct research fields (e.g. see de Hoogh et al. (2018) for air pollution estimation, Čeh et al. (2018) for real estate price estimation), it is especially interesting to consider those that fitted models with and without coordinates and compared their performance.

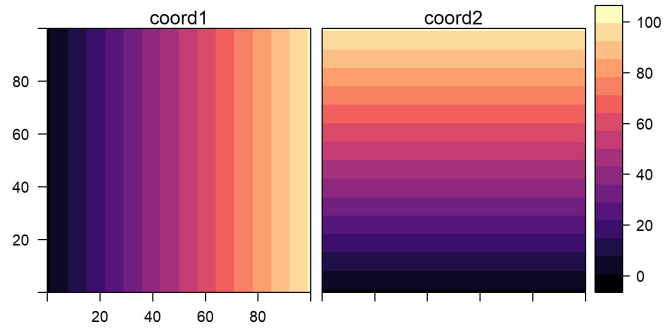


Figure 2.2: Coordinate fields in a 100x100 grid.

One of them is the study by Cracknell and Reading (2014), who tried a variety of ML-based models for geological classification mapping under different sampling scenarios. Cracknell and colleagues found that models using both environmental covariates and coordinates had higher accuracy than models using environmental variables when the number of sampling clusters was large (i.e. equivalent to random sampling). As a second example, Meyer et al. (2019) fitted regression and classification RF models with and without coordinates with highly clustered samples and, while models with coordinates appeared to be performing better when random CV was used, they had equal or larger errors once spatial CV was employed.

2.1.2.2 Euclidean Distance Fields (EDF)

As a second approach, the concept of Euclidean Distance Fields (EDF) by Behrens et al. (2018) proposes adding seven additional predictors into the ML model: two coordinate fields, four distance fields from the four corners of the study area, as well as the distance from the centre of the study area (Figure 2.3). The authors claimed that by including these distance-based fields one can account for both residual spatial autocorrelation and non-stationarity (via interaction of EDF with environmental predictors). To prove the efficacy of their model, they provided two soil composition interpolation examples in two different study areas. In both of them, the model with environmental predictors + EDF performed better (using random k-fold CV) than using environmental predictors + coordinates, or environmental predictors only.

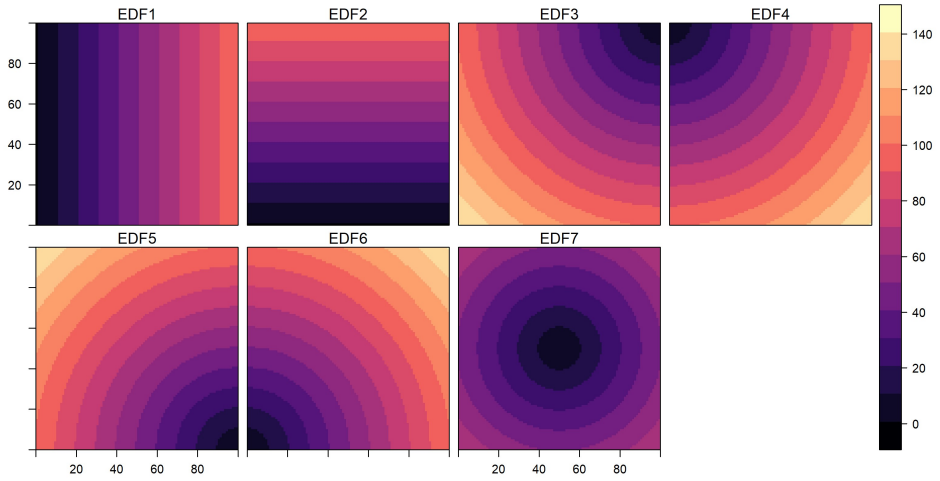


Figure 2.3: Euclidean Distance Fields in a 100x100 grid: coordinates (EDF1, EDF2), distance to study area corners (EDF3-6), and study area centre (EDF7).

2.1.2.3 Random Forest for spatial prediction (RFsp)

The concept of Random Forest for spatial prediction (RFsp) was introduced by Hengl et al. (2018). Briefly, the authors suggest a framework for predictive mapping in which (Euclidean or any other type) distances to each of the training points fields are included as n potential covariates in a RF model (Figure 2.4). The authors argue that, by incorporating distances to the rest of environmental predictors, RFsp resembles regression-kriging yet requires less expert knowledge, since no variogram estimation or fitting is needed and the trend model is handled automatically. However, they also point out that RFsp can be very computationally demanding when the number of sampling points is large, and they recommend considering quality sampling and validation methods to ensure that the model is not performing extrapolation in the predictor space or overfitting.

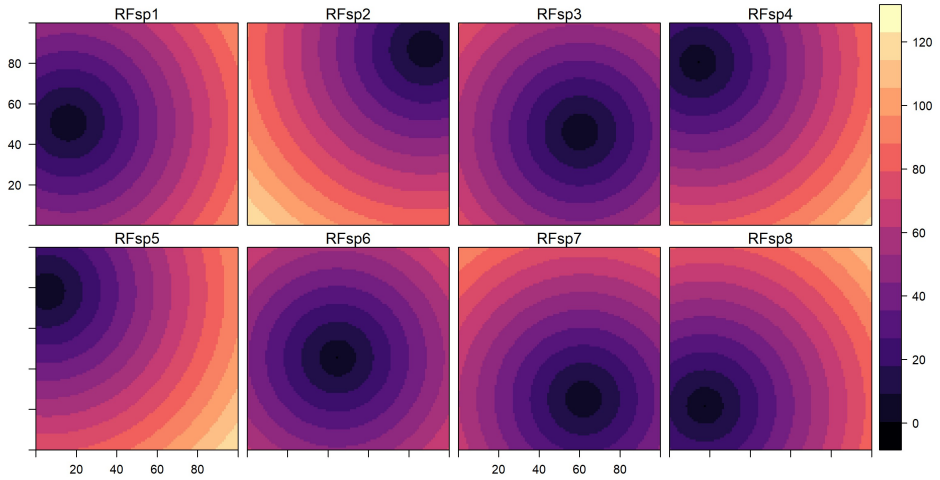


Figure 2.4: RFsp predictors in a 100x100 grid. A random selection of 8 training points has been selected for illustration purposes.

2.1.2.4 The risk of overfitting with geographic predictors

Even though explicit modelling of space in ML may solve the problem of unexplained spatial variation, it has also been identified as potentially problematic in several studies. In Meyer et al. (2019), the authors argued that including heavily autocorrelated predictors that refer to the geolocation of the samples (e.g. coordinates, Euclidean distances) or other non-causal predictors (e.g. a DEM for outcomes in which elevation is irrelevant) may lead to spatial overfitting, as the ML models fit the spatial structure of the data rather than its generating process. In the study by Meyer and colleagues, this effect became apparent when models were validated using spatially-explicit CV methods, and when artefacts in the predicted surfaces could clearly be identified.

Other authors have also pointed out this idea: Roberts et al. (2017) reflected on two common and conflicting sources of problems in spatial prediction: first, ignoring residual autocorrelation when modelling; and second, overfitting residual dependencies with non-causal predictors that share the same residual structure. Fourcade et al. (2018) were able to successfully model species distributions using pseudo-predictors derived from classical paintings. The authors argued that these predictors fitted the spatial structure rather than the underlying ecological process, leading to better performance statistics than models using climate variables when evaluated using standard CV methods. Finally, artefacts in the predicted maps when using coordinates and distances predictors in ML models have also been identified in other studies (Cracknell and Reading, 2014; Li et al., 2011).

2.2 Cross-validation methods for spatial interpolation

In order to assess the performance of a model when interpolating the outcome of interest, as well as to decide between multiple models, CV methods have been largely used in the spatial prediction literature (see supplementary table A1.1 of Roberts et al. 2017 for a systematic review). Standard CV methods to assess the generalization of a model are based on the key assumption of independence between train and test data (Hastie et al., 2009). This assumption conflicts with the nature of spatial data, which largely exhibits dependence structures (see Tobler's first law of Geography). The consequences of using random CV methods for spatial prediction problems have been widely acknowledged. To name a few recent examples, Ploton et al. (2020) found that standard CV methods for large-scale biomass mapping largely underestimated the true error of the predicted maps; Misiuk et al. (2019) used RF models to predict categorical and continuous geological outcomes and found modest decreases in performance when using spatially-explicit CV methods; while Meyer et al. (2019) found dramatic decreases of performance statistics in a Land Use and Land Cover classification (from accuracy and kappa>0.99 for random CV to $\sim$0.7 for spatial CV).

In this section, two widely used CV methods, as well as their spatial extensions suggested in the literature, will be reviewed: k-fold and Leave-One-Out CV.

2.2.1 K-fold and spatial block cross-validation

Random k-fold CV is applied by randomly partitioning the training data into k folds. For each of them, we predict the outcome fitting a model to the remaining $k-1$ folds and evaluate its prediction accuracy in the excluded fold using one or several statistics. Finally, performance measures are averaged across folds. Typical values for k are 5 and 10, being small k more computationally efficient yet more prone to bias (Kuhn and Johnson, 2013).

Spatial block CV extends the idea of k-fold CV by defining folds not randomly, but defined as blocks in the geographic space, so that points that are close in space belong to the same block (Figure 2.5). A good overview of possible spatial blocking strategies is given in Valavi et al. (2019). Many studies have used spatial blocking as a CV technique for model tuning, variable selection, and model validation (e.g. Meyer et al. 2019; Ploton et al. 2020; Roberts et al. 2017).

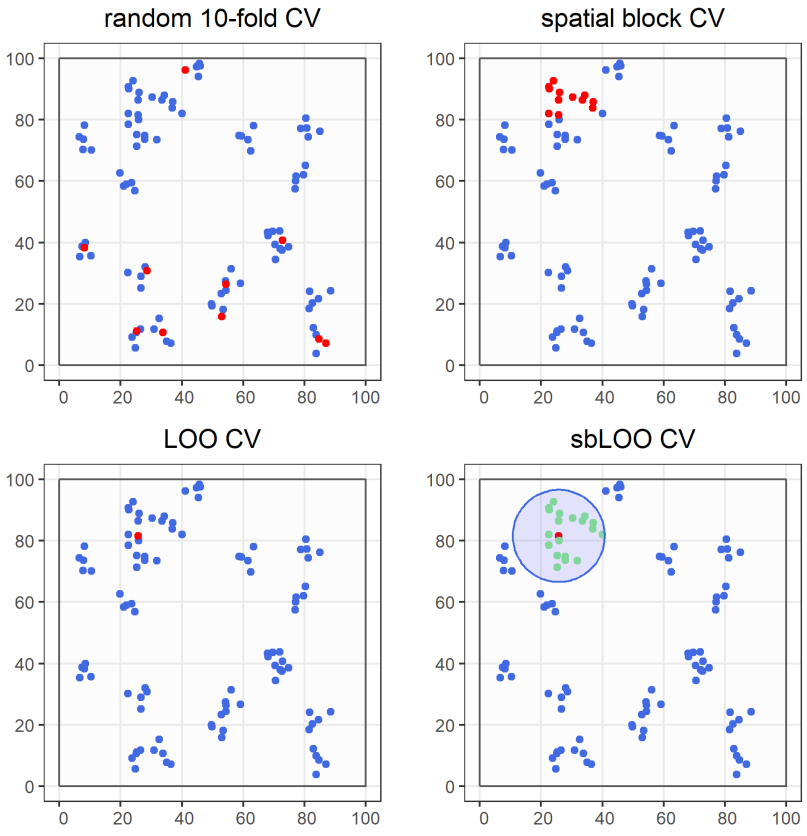


Figure 2.5: Illustration of one iteration of four different CV strategies: held-out point(s) (red), training data (blue), and exclusion buffer (circle) and left-out samples (green) in sbLOO.

2.2.2 Leave-One-Out and spatial buffer Leave-One-Out cross-validation

Leave-One-Out (LOO) CV is a specific case of k-fold CV where $k = N$, the number of observations, and therefore each observation is hold-out sequentially (Kuhn and Johnson (2013), Figure 2.5). While some research points out that LOO yields similar results than 10-fold CV but with a higher computational burden (Molinaro et al., 2005), other authors claim that LOO CV has a higher variance and a lower bias than k-fold CV (Hastie et al., 2009).

Spatial buffer Leave-One-Out (sbLOO) CV was introduced in the ecological literature (Telford and Birks, 2009) as an extension of LOO to remove dependence between training and test points in spatially structured environments in order to evaluate model transferability. Briefly, when validating each of the points, observations within a radius s are excluded from the training and validation data (Figure 2.5). This method has been used in several studies (e.g. Le Rest et al. 2014; Ploton et al. 2020; Roberts et al. 2017).

2.2.3 Finding the optimal radius/block size

The two presented spatial CV methods depend on a parameter: the radius/block size. As highlighted in the simulation study by Roberts et al. (2017), should the radius/block be too small, it could lead to error underestimation, while if the opposite is true it may lead to overestimation due to unnecessary extrapolation in the predictor space. Three different methods have been suggested to choose it:

1. Residual autocorrelation range: The most common suggestion found in the literature is to choose a radius equal to the range of the semivariogram estimated on the model residuals (e.g. Telford and Birks 2009; Trachsel and Telford 2016).
2. Outcome autocorrelation range: Roberts et al. (2017) argued that the radius should be equal to the autocorrelation range of the outcome, as measuring it on the residuals may hide overfitting that might have occurred in the modelling stage.
3. Landscape autocorrelation range: Valavi et al. (2019) suggested using the median autocorrelation range of the candidate predictors in the model (via semivariogram estimation using random sampling on the rasters) as a measure of the radius size.

Contrasting to our literature review and as already mentioned in the introduction section 1.5, we think that the suggested methods to estimate the spatial CV parameters, which aim to achieve independence between train and test data, may not be appropriate for geographic interpolation problems, where at least a subset of the prediction pixels will not be independent of the training data (e.g. pixels neighbouring a training point), and thus error estimated with these radii/block sizes may be overestimated. Surprisingly, we did not find any similar reflection in the body of literature that we reviewed.

2.3 Factors affecting ML modelling and validation strategies

Three of the most important factors determining whether spatially-explicit modelling and validation are appropriate are: 1) Objectives of the prediction 2) Spatial autocorrelation of the landscape, 3) Sampling design. To evaluate the effect of these factors, scientists have mostly turned to simulation analyses, since they allow to reproduce a wide variety of scenarios that would be very challenging to cover with real data. Overall, the evidence for ML-based modelling and validation strategies, including those spatially-explicit described in sections 2.1.2 and 2.2, is limited and fragmented. A summary of these findings is found in this section.

2.3.1 Prediction objectives: interpolation vs. extrapolation

The easiest factor to determine, yet possibly the one that can have the largest impact when choosing the modelling and validation methods, is the objective of the prediction: geographic interpolation vs. extrapolation (Li and Heap, 2014). The impact of the prediction objectives when using spatially-explicit vs. non-spatial models is elucidated in Rocha et al. (2019). Rocha and colleagues designed a simulation study which compared the performance of a simple linear spatial model and complex non-spatial ML models in 1) Test sites in the study area (geographic interpolation) and 2) a new landscape realization (geographic extrapolation). They found out that, while the simple linear spatial model performed the best for test sites, it was the worst choice for new realizations of the landscape. Therefore, the authors concluded that spatial models are not generalizable to completely new locations as spatial structures would probably not be shared.

The objectives of the prediction may also impact model validation methods. Roberts et al. (2017) stated that random CV can be a reasonable choice if estimates of the prediction error in locations of the same geographic and predictor space than the training data are required, while if either extrapolation in the predictor/geographic space is desired, blocking in the predictor/ geographic space is needed. The authors support this statement with a simulation study showing prediction error underestimation when using random CV to assess prediction at an independent location.

2.3.2 Spatial autocorrelation range

In the last years, Rocha and colleagues have studied the effect of the spatial autocorrelation of a landscape on spatial modelling and validation strategies via simulation analyses. In their first study (Rocha et al., 2018), a wide range of ML models (support vector machines, partial least squares regression, linear models and variations thereof) were fitted for Leave Area Index prediction using simulated hyperspectral data cubes with different spatial autocorrelation ranges.

Amongst other results, they found that random CV errors for medium and large autocorrelation ranges were smaller than errors on an independent data set, whereas for small ranges they were similar. This suggested an underestimation of random CV errors in highly autocorrelated landscapes.

The second study by Rocha et al. (2019) used the same design to prove that non-spatial interpolation ML models performed worse than simpler, bayesian spatially-explicit linear models when the spatial autocorrelation ranges were large (they were roughly equivalent for short ranges).

2.3.3 Sampling design

Wadoux et al. (2019) investigated the optimal sample design for RF-based soil sciences mapping applications. They concluded that methods sampling according to the feature space will generally obtain better results than those sampling in the geographic space, which would be more suited to univariate interpolation methods such as Ordinary Kriging. However, in many cases the researcher must use the already existing samples (e.g. monitoring stations, existing samples) whose locations are not optimised for predictive mapping. In those cases, the danger of overfitting under clustered sampling designs when using highly autocorrelated predictors such as geographic coordinates and/or Euclidean distances has been acknowledged by different authors such as Hengl et al. (2018); Meyer et al. (2019).

One of the few studies examining the effect of sampling on spatial interpolation modelling and validation is that of Rocha et al. (2020). Rocha and colleagues use the same simulation framework as their 2018 and 2019 studies described in subsection 2.3.2 to prove that, among the included sampling designs (random, systematic, lattice close pair, and lattice in-fill), systematic sampling yielded the lowest independent test data errors regardless of the model, whereas close pair and in-fill designs delivered the highest. Interestingly, they found reverse patterns when evaluating CV results, i.e. systematic designs had the highest CV errors.

Another relevant study is that of Cracknell and Reading (2014) which, as already described in section 2.1.2.1, found that models using both environmental covariates and coordinates had higher accuracy than models using environmental variables only when the number of sampling clusters was very large and hence roughly equivalent to random sampling, whereas performances of the two models were similar for a small number of clusters.

Chapter 3

Methods

3.1 Spatial prediction sandbox overview

The spatial prediction sandbox is a simulation framework that allows to simulate predictor and outcome fields according to a certain degree of spatial autocorrelation, and samples with different number of points and distributions. With these data, we can test different standard and spatially-explicit ML models and evaluate them using different standard and spatially-explicit validation strategies, which will allow us to answer the research questions formulated in this Master Thesis. Figure 3.1 shows an overview of the architecture and methods of the sandbox:

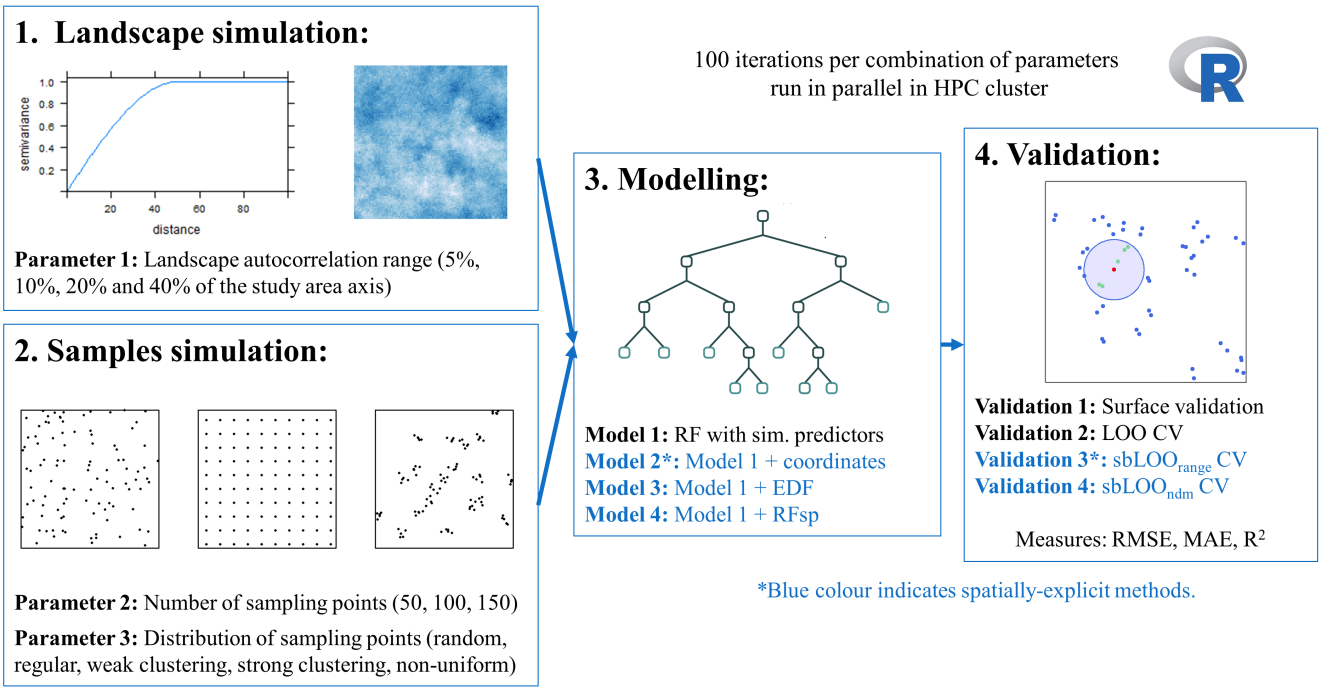


Figure 3.1: Spatial prediction sandbox architecture, methods, and parameters overview.

Briefly, the sandbox evaluates 4 landscape configurations (parameter 1) · 15 sampling configurations (parameters 2 and 3), resulting in 60 possible scenarios for analysis. For each of them, 4 different models are analysed, thus yielding possible 240 model-scenario combinations to evaluate. For each model under each scenario, 4 different validation strategies and 3 different evaluation measures are used, thus resulting in 2,880 possible metrics. 100 iterations of the

sandbox were run in order to obtain a distribution of the 2,880 evaluation metrics.

In the following sections, methods of each of the spatial prediction sandbox blocks are explained in detail, together with examples of intermediate outputs generated in one iteration. Finally, methods used to extract information and draw conclusions from the simulations results, as well as the computational implementation of the sandbox, are described. The sandbox code is available and documented at the github repository: https://github.com/carlesmila/spatial-prediction-sandbox.

3.2 Block 1: Landscape simulation

3.2.1 Methodology

After defining the study area as a 100x100 square grid (i.e. 10^4 grid cells), the first step is to simulate the landscapes consisting of covariate and outcome fields. To do so, we adapted simulation framework 2 from Van der Laan et al. (2007). Briefly, it consisted in a simulation of 20 i.i.d. random variables X_i where $X_i \sim N(0, 16)$, out of which the following outcome Y is generated with the equation:

$$Y = X_1X_2 + X_{10}^2 - X_3X_{17} - X_{15}X_4 + X_9X_5 + X_{19} - X_{20}^2 + X_9X_8 + \varepsilon$$

It can be observed that additive and multiplicative terms, as well as non-linearities, are included. The 8 variables not used in this equation remain as potential candidate predictors in the models. The noise ε was also distributed as $\varepsilon \sim N(0, 16)$.

We adapted this framework to a spatially structured environment as follows:

1. Covariates are 2-dimensional stationary and isotropic Gaussian random fields subject to spatial autocorrelation, expressed with spherical semivariograms:

$$\gamma(h) = \begin{cases} \tau^2 + \sigma^2(\frac{3h}{2\phi} - \frac{1}{2}(\frac{h}{\phi})^3) & \text{if } 0 \leq h \leq \phi \\ \tau^2 + \sigma^2 & \text{if } h > \phi \end{cases}$$

   where h is a distance, $\tau^2 = 0$ is the nugget effect, $\sigma^2 = 1$ is the partial sill, and ϕ is the range (simulation parameter). Substituting by those values, we have the semivariogram:

$$\gamma(h) = \begin{cases} \frac{3h}{2\phi} - \frac{1}{2}(\frac{h}{\phi})^3 & \text{if } 0 \leq h \leq \phi \\ 1 & \text{if } h > \phi \end{cases}$$

2. The error term ε includes, in addition to a standard normal noise term (ε_1), a 2-dimensional Gaussian random field error term ($\varepsilon_2(s)$, where s is a location in the study area) subject to spatial autocorrelation according to the same semivariogram used for the covariates:

$$\varepsilon(s) = \varepsilon_1 + \varepsilon_2(s) \text{ where } \varepsilon_1 \sim N(0, 1) \text{ and } \varepsilon_2|\theta \sim N(0, \Sigma(\theta))$$

3. We lowered the variance of the predictors to 1 (expressed as $\lim_{h\to\infty}\gamma(h)$, i.e. the sill) to lower the signal-to-noise ratio of the simulation to a reasonable amount (signal-to-noise ratio = $\frac{\mathrm{Var}(\mathrm{E}(Y|X))}{\mathrm{Var}(Y-\mathrm{E}(Y|X))} = \frac{\mathrm{Var}(f(X))}{\mathrm{Var}(\varepsilon)} = 5$).

In order to simulate the random fields, we used the Sequential Gaussian simulation algorithm for unconditional simulation (i.e. purely based on the semivariogram defined *a priori* with no conditioning observed data) implemented in the R package `gstat` (Pebesma, 2004). As described in Gebbers and de Bruin (2010), sequential simulation visits all prediction locations by following a randomly ordered path. For each node $\mathbf{x}_i$:

1. Use simple kriging with the given semivariogram to find the kriging estimate $\hat{Z}(\mathbf{x}_i)$ and its variance $\hat{\sigma}^2_K(\mathbf{x}_i)$ using previously simulated values, which can be limited to a search radius or a number of nearest neighbours (we set `nmax` argument to 100 nearest neighbours to speed up computations).
2. Define a Gaussian conditional cumulative distribution function (ccdf) based on the kriging estimate (μ) and variance (σ^2) from the previous step.
3. Draw a pseudo-random value from the ccdf and assign it to the location.
4. Go to the next stop and start from top until done.

For the autocorrelation range parameter, we defined four possible values: a 5%, 10%, 20% and 40% of the grid axis size, i.e. 5, 10, 20, and 40 units given a grid of 100x100 (Figure 3.2).

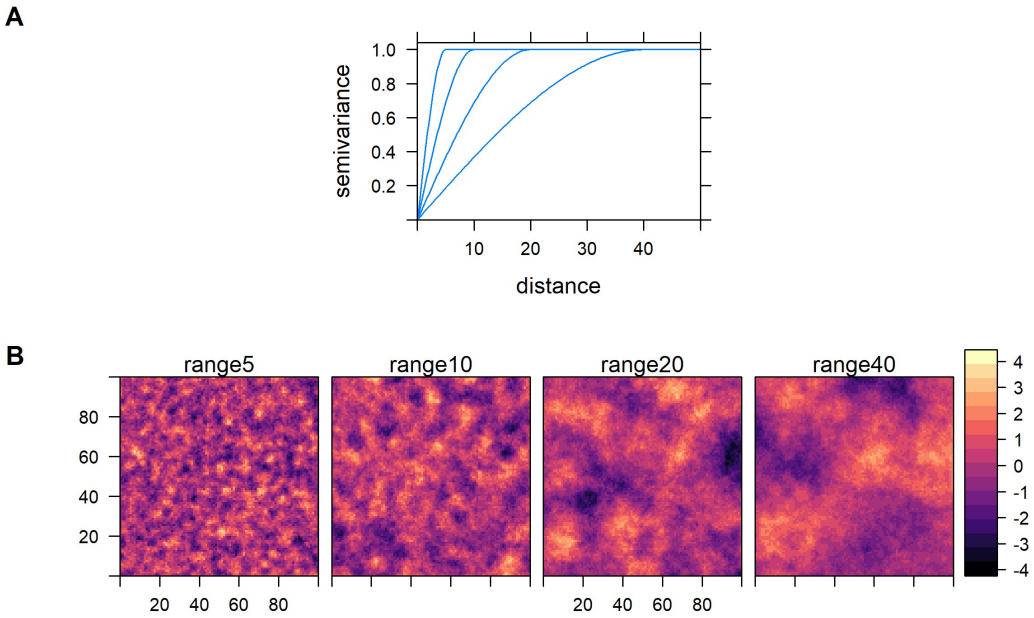


Figure 3.2: Semivariograms used in the sandbox (A) and one example simulation realization of each of them (B).

3.2.2 Workflow example

The user calls the sandbox with the default parameters and only one user-required parameter: the landscape autocorrelation range, which we set to 10% for this example. The sandbox is initiated by defining the study area polygon and grid (100x100) in both vector (points) and raster format. Next, 20 different Gaussian different fields are generated according to the semivariogram range parameter (Figure 3.3).

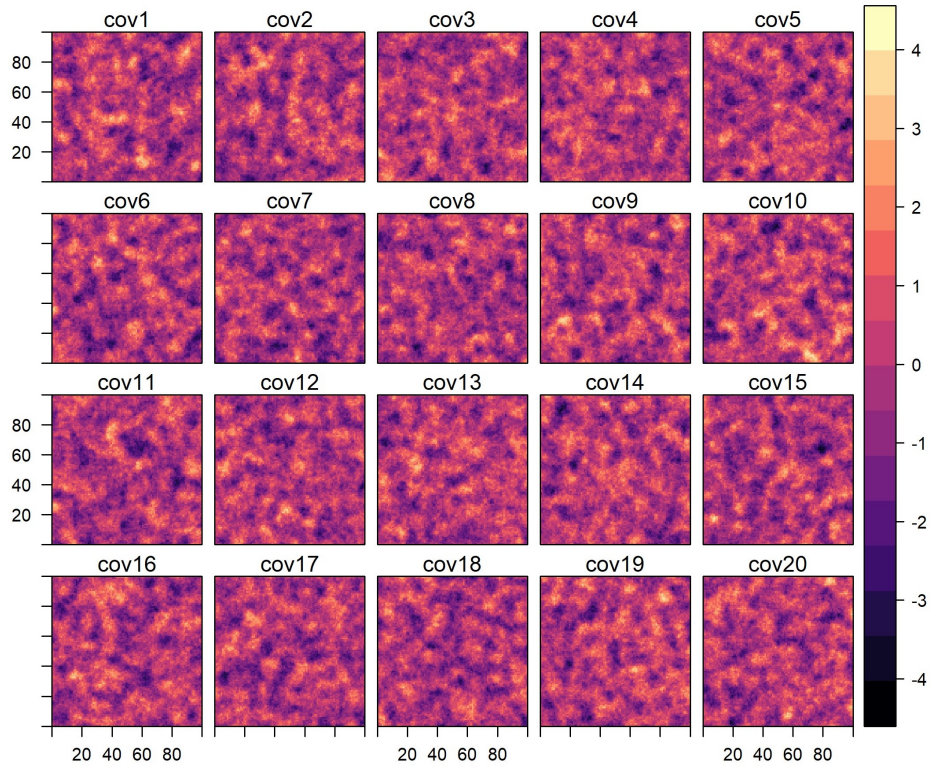


Figure 3.3: One simulation realisation of the 20 covariate random fields (range: 10%).

From the simulated covariates, the response is derived using Van der Laan et al. (2007) aforementioned formula, and random and spatially autocorrelated noise are added to obtain the final outcome field (Figure 3.4). With this, the first sandbox block is concluded.

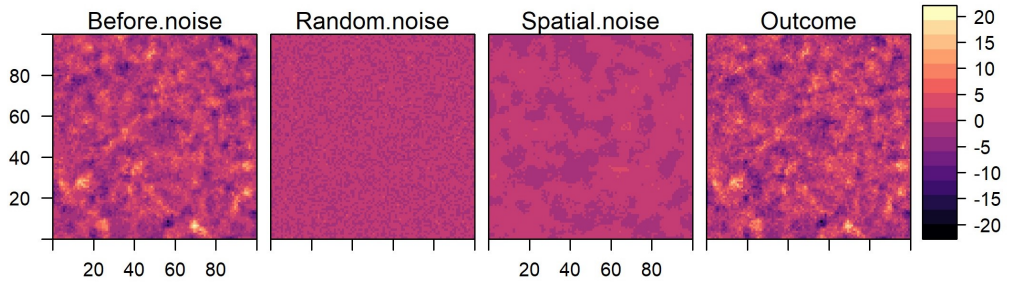


Figure 3.4: Example of outcome derived from covariates, simulated random and spatially correlated noise fields, and final outcome with noise added.

3.3 Block 2: Samples simulation

3.3.1 Methodology

The second block simulates the spatial locations of the samples. This depends on two different parameters: the number of sampling points and their spatial distribution. By default, we defined three different values for sampling points (50, 100, and 150) and five different possible distributions, from which we simulated using the original version and variations of the function `st_sample` included in the package `sf` as follows:

1. Random: A set of points is simulated using two uniform distributions for the two dimensions with parameters equal to the boundaries of the study area, i.e. $X \sim U(0, 100)$, $Y \sim U(0, 100)$, which are bound to form coordinate couples.
2. Regular: A regular grid is formed based on the number of points to simulate and the dimensions of the study area.
3. Weak clustering (clust1): We first simulate $n/5$ parent random points in the study area. We compute 5-units buffers and for each of them and we randomly sample an additional 4 offspring points per buffer, yielding a total of $n \cdot 4/5$ offspring points.
4. Strong clustering (clust2): Similar to clust1, but simulating $n/10$ parents and 9 offspring points per parent (i.e. $n \cdot 9/10$ total offspring).
5. Non-uniform: We divided the study area into 5x5 squares of side length 20, and selected 5 of those randomly (Figure 3.5). Then, we sampled random points within the selected squares. Note that this method is different than preferential sampling, where the samples and the process we want to model are dependent, since in this case we define sampling areas randomly and *a priori* (Diggle et al., 2010).

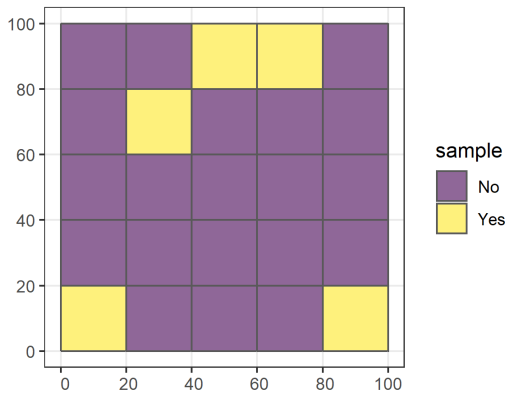


Figure 3.5: Example of a partition of the study area to perform non-uniform sampling.

As a result, 15 different point samples are simulated for each landscape by default (all combinations of 50/100/150 points with a random/regular/clust1/clust2/non-uniform distribution).

3.3.2 Workflow example

In this example, 5 different sets of 100 points according to the 5 different distributions considered in the sandbox are simulated (Figure 3.6).

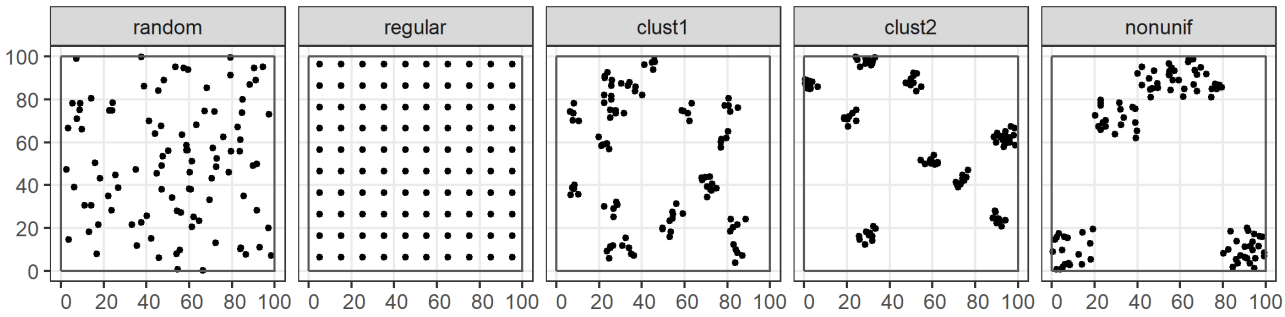


Figure 3.6: One simulation realisation of 100 points according to the 5 sampling distributions considered in the spatial prediction sandbox.

3.4 Block 3: Modelling

3.4.1 Methodology

Four different strategies for modelling are used in the sandbox: a non-spatial RF model and three spatially-explicit methods already covered in detail in section 2.1.2:

- Benchmark model: Non-spatial model using the 20 simulated covariate random fields as predictors of the outcome.
- Coordinates: Spatial model using the 20 simulated fields + 2 coordinate fields.
- EDF: Spatial model using the 20 simulated fields + EDF (2 coordinate fields, Euclidean distance to 4 corners, distance to centre).
- RFsp: Spatial model using the 20 simulated fields + sample distance fields (n Euclidean distance fields to each of the sample training points).

In order to shorten computation times, we did not perform hyperparameter tuning of the models, but rather took default values recommended for RF regression. Those were a number of trees (`ntree`) equal to 500, a minimum node size (`nodesize`) equal to 5, and a number of predictors to be sampled in each tree (`mtry`) equal to $1/3 \cdot k$ except for the RFsp model where, following author's recommendations (Hengl et al., 2018), was set to $2/3 \cdot k$. Model fitting was done using the `randomForest` function encapsulated in the ML `caret` package.

3.4.2 Workflow example

An example of the model fitting block will be shown together with model validation in section 3.5.2.

3.5 Block 4: Validation

3.5.1 Methodology

Each of the models is validated using four different strategies:

1. Surface validation: The model is used to predict the whole surface of the 100x100 grid. Since we know the actual value of the outcome at each pixel, we base our validation on a comparison of all grid cells, thus giving us the "true" performance of the model.
2. Leave-One-Out (LOO) CV: explained in section 2.2.2 of the literature review.
3. Spatial buffer LOO CV, radius based on range ($\text{sbLOO}_{\text{range}}$): the sbLOO method was explained in section 2.2.2 of the literature review. Out of the three strategies suggested in the literature for choosing the radius of sbLOO (section 2.2.3), we used the median autocorrelation range of the 20 predictors (Valavi et al., 2019): a 10% of each of the covariates raster grid cells (i.e. 1000 cells) were randomly sampled and a spherical semivariogram (coercing the nugget to 0 to help with fitting) was fitted automatically using the function `autofitVariogram` of the package `automap`.
4. Spatial buffer LOO CV, radius based on Nearest Distance Matching ($\text{sbLOO}_{\text{ndm}}$): We propose a new method called Nearest Distance Matching (NDM) to choose the sbLOO radius for spatial interpolation problems based not only on the landscape autocorrelation, but also on the distribution of the samples. Briefly, we used point process methods to find radii that make the distribution of the nearest distances between the held-out and train data in sbLOO CV to resemble as much as possible the distribution of the nearest distances between grid cells for which we want to predict and all training data, for distances in which autocorrelation was present. Our new suggested method is thoroughly described in Appendix A, where the theoretical background and worked examples are given.

For both sbLOO strategies, we tested whether in each CV iteration at least a 50% of the observations were available for the model fit, and if not, we decreased the radius by 1 unit until the condition was met. For each of the four validation strategies, we used three widely used statistics in the ML literature for regression problems: the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the R^2 between the actual and predicted values.

3.5.2 Workflow example

Here we continue the step-by-step workflow by looking at model fitting and validation of the benchmark model in the case of autocorrelation range of 10% and 100 random samples. Before we start modelling, we calculate the median landscape autocorrelation range of the 20 predictors by randomly sampling a 10% of the raster pixels and fitting a semivariogram automatically.

The estimated covariate median range serves both as the radius in $\text{sbLOO}_{\text{range}}$, as well as a parameter for the NDM algorithm (Figure 3.7).

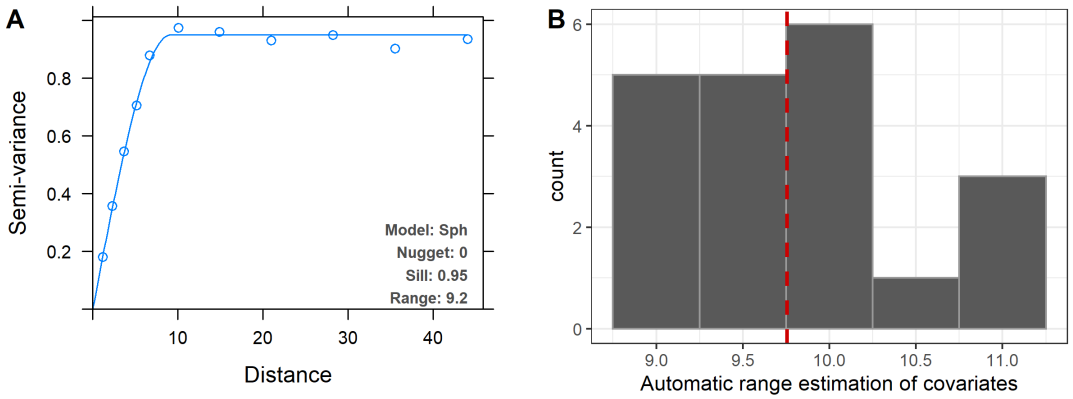


Figure 3.7: Example of an autofitted semivariogram to one of the covariates (A) and histogram of the 20 estimated ranges and their median value (red dashed line) (B)

We obtained a median range of 9.76, which is aligned with the simulation parameter of range=10% we set in this example. Now, we can apply the NDM algorithm described in Appendix A to estimate the appropriate radius for $\text{sbLOO}_{\text{ndm}}$. The resulting radius is 0, which makes $\text{sbLOO}_{\text{ndm}}$ equivalent to LOO.

Next, we fit the ML model (benchmark RF model), which we can use to predict the continuous surface. We can use the outcome and predicted surfaces to derive an error surface as the difference between the two, which will be subsequently used to derive surface RMSE and MAE statistics; as well as to assess the correlation between the two via scatterplot and linear model fit, which is the basis for the R^2 (Figure 3.8).

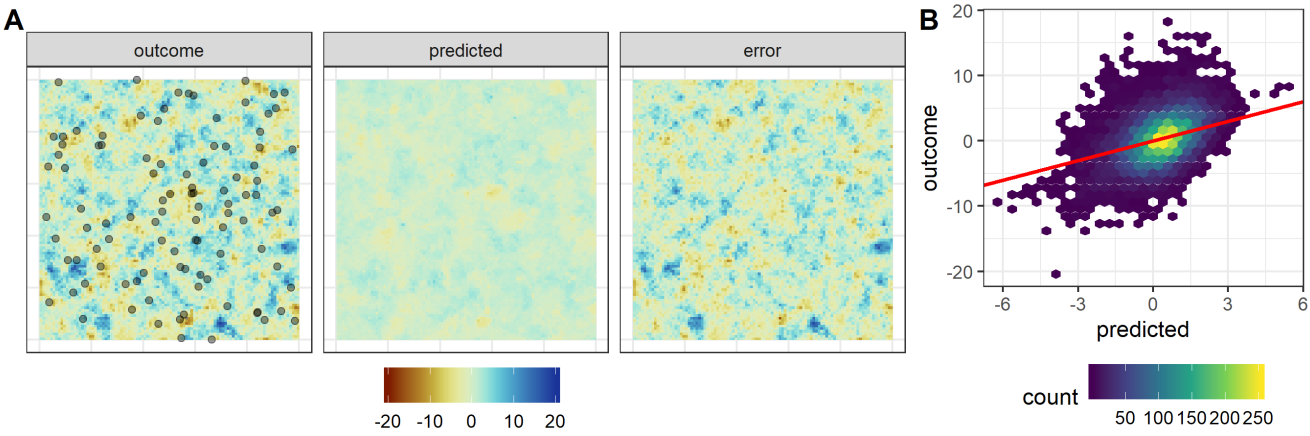


Figure 3.8: Example of outcome, predicted, and error surfaces (A) and hexagon-binned scatterplot (one observation per pixel) with $y = x$ line (red) (B).

Finally, we further validate models by running 1) LOO CV 2) $\text{sbLOO}_{\text{range}}$ CV and 3) $\text{sbLOO}_{\text{ndm}}$ CV and assessing, for each of them, the RMSE, MAE, and R^2 (table 3.1). With the model

validation, the iteration of the sandbox concludes.

Statistic	surface	LOO	$\text{sbLOO}_{\text{range}}$	$\text{sbLOO}_{\text{ndm}}$
RMSE	3.16	3.38	3.49	3.38
MAE	2.42	2.60	2.69	2.59
R^2	0.19	0.13	0.07	0.13

Table 3.1: Example of validation statistics for one iteration of the sandbox.

3.6 Analysis of the results

We analysed the results of sandbox simulations in two blocks according to our two research questions: modelling and validation. Modelling results were analysed by plotting surface validation metrics (RMSE, MAE, and R^2) comparing the whole predicted surfaces of each of the models to their actual outcome surfaces by sample size, distribution, and range; we also computed summary statistics (mean, standard deviation (SD)) to perform some numerical comparisons. Furthermore, we plotted some examples of predicted surfaces for interpretation purposes.

For the validation block, we computed ratios of the CV statistics (RMSE, MAE) by dividing them by their actual surface error, i.e. $\frac{\text{LOO}}{\text{surface}}$, $\frac{\text{sbLOO}_{\text{range}}}{\text{surface}}$, and $\frac{\text{sbLOO}_{\text{ndm}}}{\text{surface}}$. For R^2 analyses we subtracted $R^2_{\text{LOO}} - R^2_{\text{surface}}$, $R^2_{\text{sbLOO}_{\text{range}}} - R^2_{\text{surface}}$, and $R^2_{\text{sbLOO}_{\text{ndm}}} - R^2_{\text{surface}}$ to maintain the R^2 unit interpretability. We plotted the distribution of these measures by sample size, distribution, range, and model using boxplots; and computed some summary statistics.

3.7 Implementation and parallelization

The sandbox was written in R3.6 (R Core Team, 2019) and was structured in an R project within Rstudio, available at the following github repository where all the code is available and documented: https://github.com/carlesmila/spatial-prediction-sandbox. Figures were generated in R markdown documents (also available in the repository), thus allowing for easy-to-implement results updates. The sandbox code was written to allow a certain degree of customization of the parameters of the simulation, being the outcomes described in this Master Thesis the result of applying the defaults. The arguments in the `spatialpred_sandbox` function are:

- Grid axis dimensions (`dimgrid`). Default: 100.
- Autocorrelation range of the landscape (`range`). No default, user-required parameter. Ranges equal to 5, 10, 20 and 40 were used in this thesis.
- Number of samples (`n_train`). Default: 50, 100, and 150.
- Distribution of the samples (`sample_dist`). Default: regular, random, clust1, clust2, nonunif. A subset of these can be selected.
- Models to be fitted (`models`): benchmark, coords, EDF, RFsp. A subset can be selected.

Since a total of 100 iterations per possible scenario were done, the code was organized to be run in parallel in the Palma II High Performing Computing (HPC) cluster of the University of Münster, by assigning one iteration to each core. However, unsolvable problems within the HPC cluster regarding 1) hyperthreading management and 2) random seed management made it impossible. Therefore, parallelization was limited to CV in the modelling stage of the sandbox, which fortunately was the most computationally demanding part of the process. The computation of the 100 iterations of the sandbox for all parameter combinations took approximately 96 hours to complete using one node with 72 cores and 64GB of memory.

A wide range of R packages were used in the sandbox and/or the graphics displaying the results: `doParallel` (Corporation and Weston, 2019) for parallel computing; `sf` (Pebesma, 2018) and `raster` (Hijmans, 2019) for vector and raster data management, respectively; `gstat` (Pebesma, 2004) for random field simulation and `automap` (Hiemstra et al., 2008) for automatic semivariogram fitting; `spatstat` (Baddeley and Turner, 2005) for point process management and analysis; `tidyverse` (Wickham, 2017) for data management and plotting. Other packages were used to execute other minor tasks.

Chapter 4

Results

4.1 Modelling

In order to assess our first research question (under which spatial autocorrelation and sampling conditions are spatially-explicit ML models appropriate?), we examined the distribution of validation measures (RMSE, MAE, and R^2) comparing the predicted surfaces of each of the models and the actual outcome surface by sample size, distribution, and range.

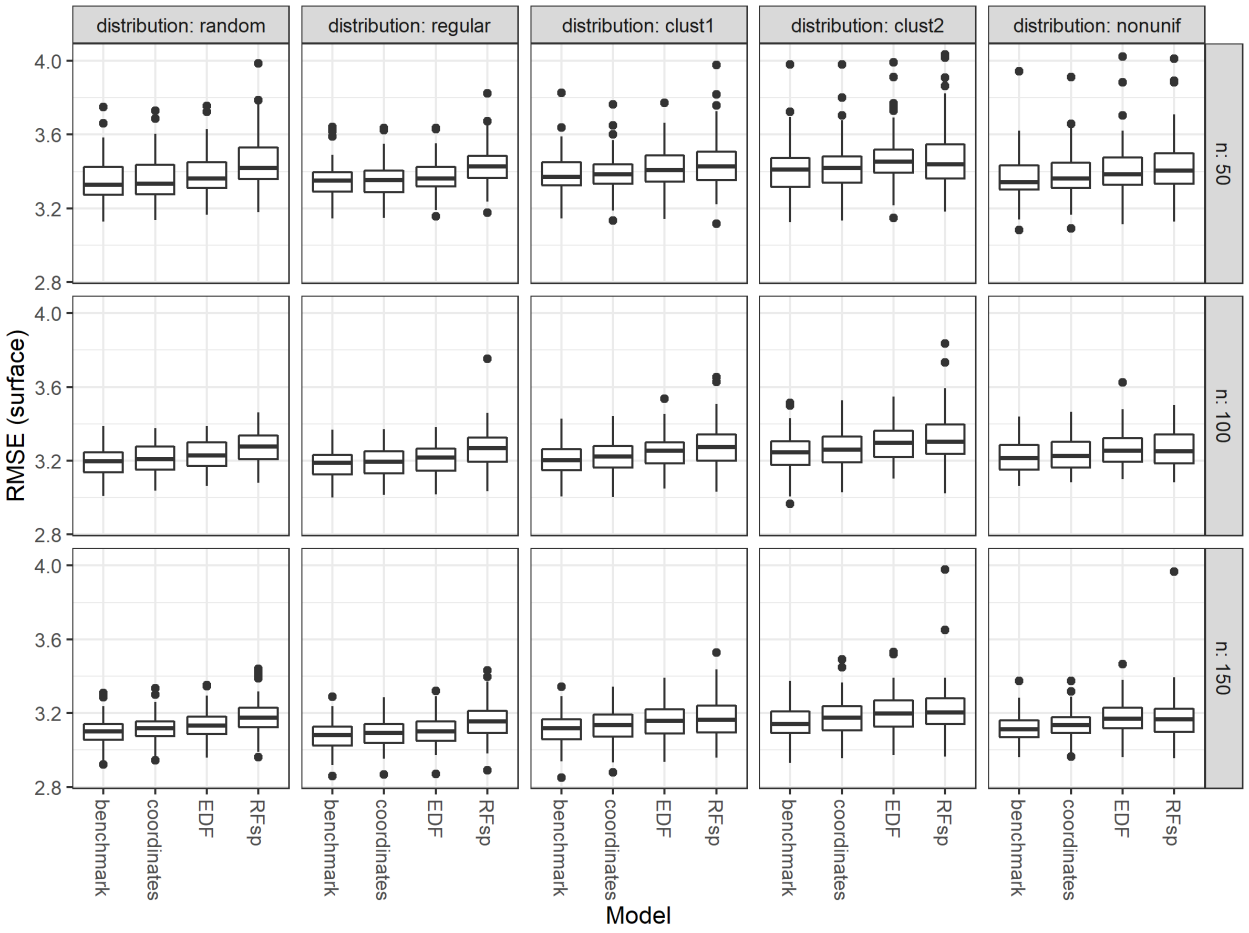


Figure 4.1: RMSE (surface) of ML spatial interpolation models by sampling distribution and size, for low spatial autocorrelation range (5%). Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

A visual inspection of the distribution of the RMSE of the predicted vs. the actual outcome

surfaces for a small range (5% of the study area) indicated a lower performance of spatially-explicit models (coordinates, EDF, and RFsp) regardless of the sample size and distribution (Figure 4.1). For example, for $n = 100$ and a clust1 design, the mean RMSE (SD) was 3.21 (0.09) for the benchmark model and 3.28 (0.12) for RFsp. Amongst spatially-explicit models, RFsp tended to be the worst choice for most designs (e.g. for $n = 50$ and a random design the mean RMSE (SD) was 3.43 (0.14) for RFsp and 3.36 (0.11) for the coordinates model).

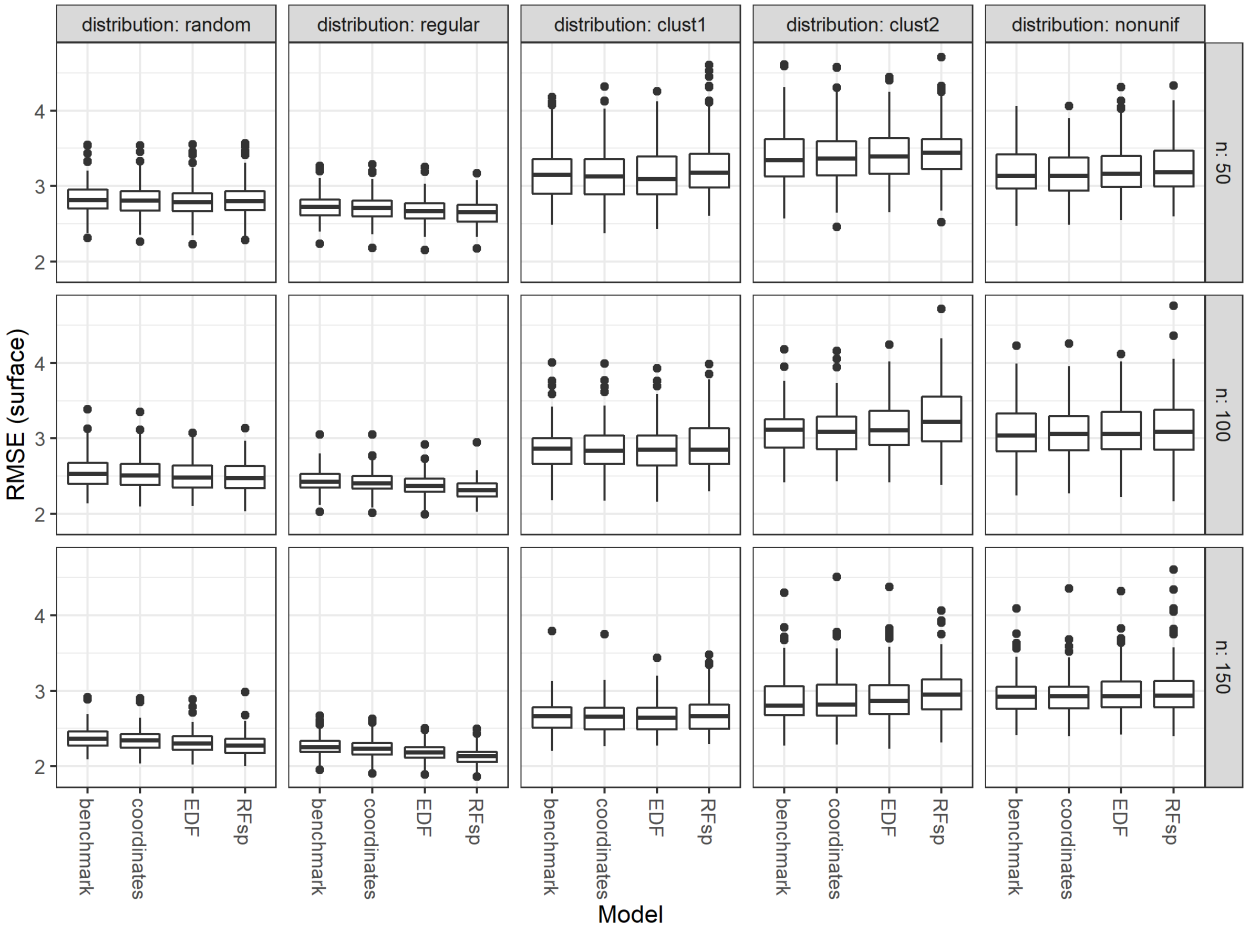


Figure 4.2: RMSE (surface) of ML spatial interpolation models by sampling distribution and size, for high spatial autocorrelation range (40%). Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

Results for the high autocorrelation scenario (range=40%) were more complex (Figure 4.2)[1]. For random and regular designs, spatially-explicit models performed better than non-spatial benchmark models, with RFsp generally being the model yielding a smaller error. For example, for $n = 50$ and a regular design, mean RMSE (SD) was 2.73 (0.19) for the benchmark model, 2.71 (0.19) for the coordinates model, 2.67 (0.18) for the EDF model and 2.65 (0.18) for the RFsp model. For clustered and non-uniform designs, all models had a similar mean RMSE (SD)

[1]One observation of RMSE=5.77 for RFsp and 50 non-uniform sampling points was removed from the graph for visualization purposes.

(e.g. for $n = 100$ and clust1 distribution 2.87 (0.3) for benchmark, 2.86 (0.31) for coordinates, 2.87 (0.31) for EDF, and 2.91 (0.34) for RFsp), except for RFsp in the strongly clustered design, whose errors were larger. The sample size did not have any effect on the results other than an overall better performance of the models with larger sample sizes.

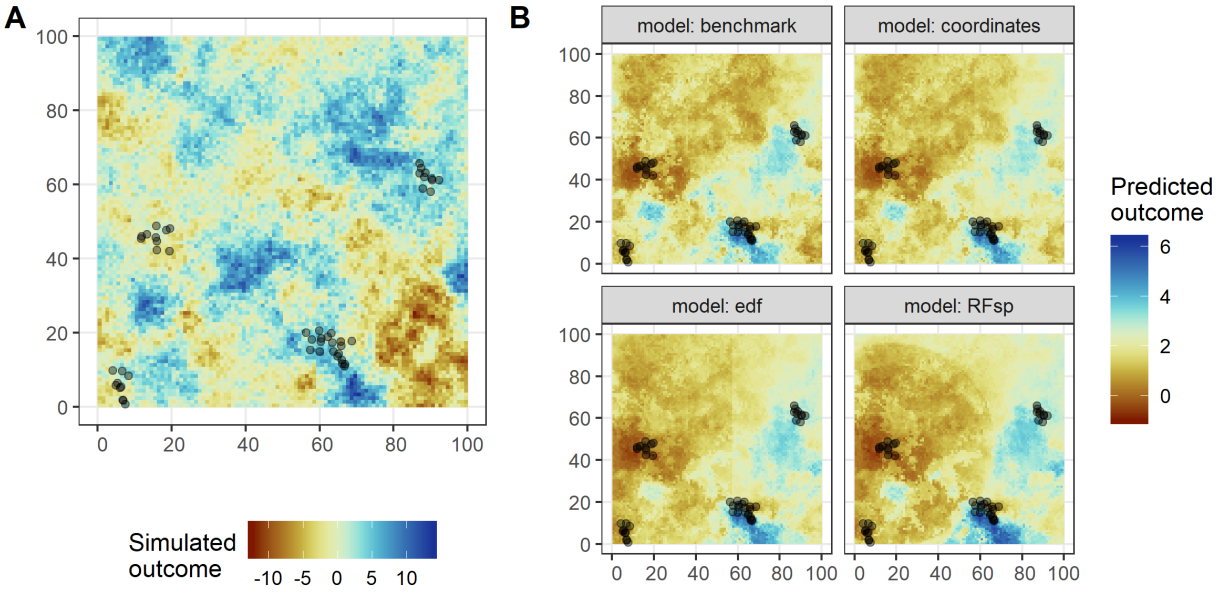


Figure 4.3: Example of a simulated outcome surface (A) and four predicted surfaces according to the different included models (B) for $n = 50$, clust2 distribution, and a landscape with range equal to 40%.

A visual inspection of a selection of predicted surfaces under each model and scenario revealed obvious artefacts for long spatial autocorrelation ranges and clustered or non-uniform sampling designs (Figure 4.3). For instance, vertical breaks in the predictions could be observed in the coordinates and EDF models possibly due to the x predictor, while RFsp clearly showed spherical patterns corresponding to Euclidean distance fields. Though not so often, artefacts were also observed for other sample distributions (e.g. figure B.1 for a regular sampling distribution).

RMSE results for all the studied autocorrelation ranges (i.e. 5%, 10%, 20%, and 40%; $n = 100$; Supplementary Figure B.2) indicated very similar results for ranges equal to 10% compared to 5%, as well as for range 20% compared to range 40%. Results for the MAE statistic and sample size $n = 100$ (supplementary Figure B.3) followed the same patterns and yielded the same conclusions as results for RMSE. Results for R^2 and $n = 100$ were also fairly similar to RMSE (supplementary Figure B.4), although the gains of using spatially-explicit models in random and regular designs and long ranges were minimal according to this validation statistic.

4.2 Validation

We assessed our second and third research questions by examining the ratio of the CV RMSE and MAE to their actual surface error, i.e. $\frac{\text{LOO}}{\text{surface}}$, $\frac{\text{sbLOO}_{\text{range}}}{\text{surface}}$, and $\frac{\text{sbLOO}_{\text{ndm}}}{\text{surface}}$. Note that a ratio < 1 would mean the CV is underestimating the actual error of the interpolated map, while a ratio > 1 would mean overestimation. For R^2 analyses we subtracted $R^2_{\text{CV}} - R^2_{\text{surface}}$ to maintain the interpretability of the units of the statistic.

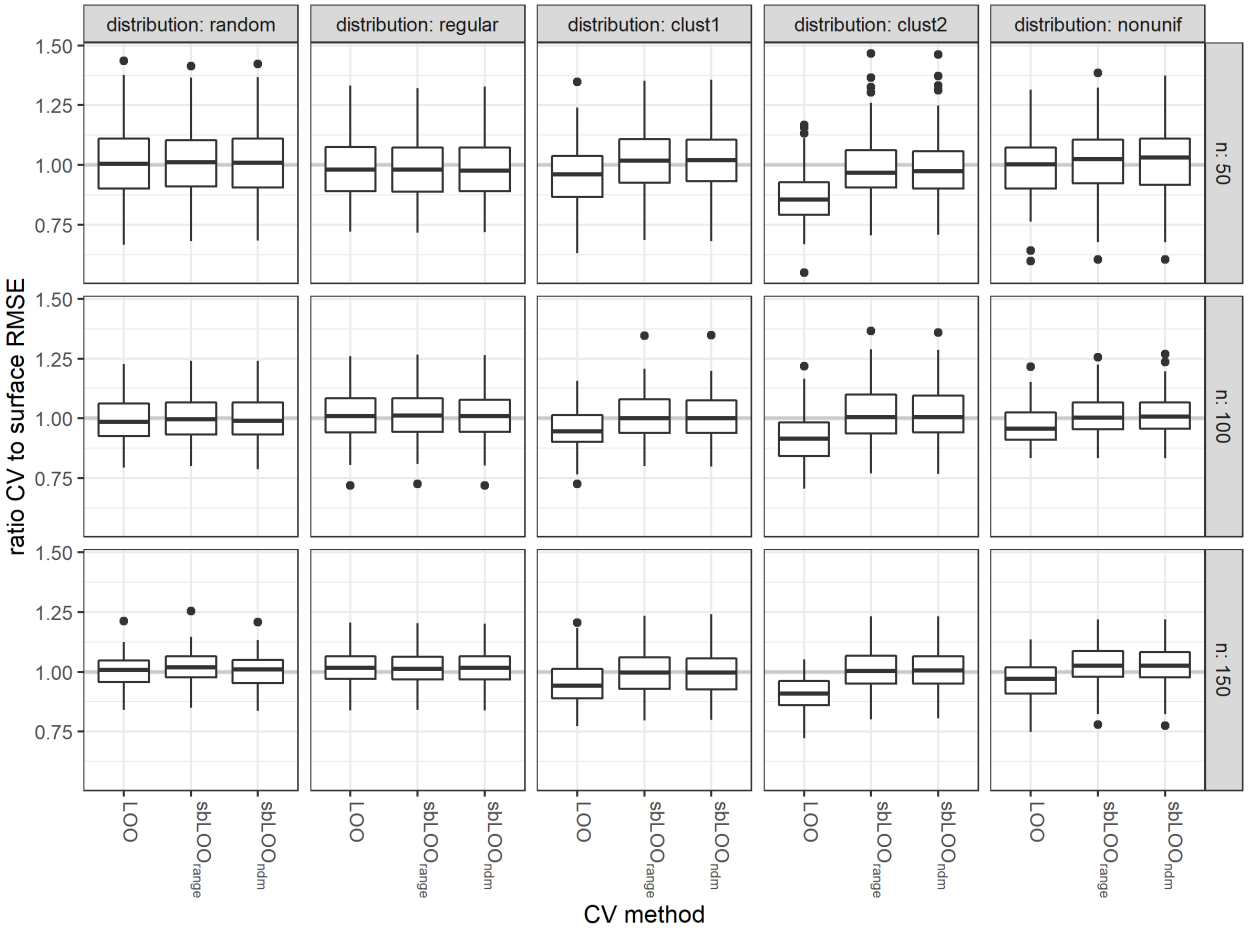


Figure 4.4: Ratio of CV to surface RMSE of benchmark models by sampling distribution and size, for low spatial autocorrelation range (5%). Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

Results for low spatial autocorrelation and the RMSE validation statistic (Figure 4.4) showed that, on average, all CV methods correctly estimated the actual error of the surfaces of the benchmark model under a random and regular design, regardless of the number of sampling points (e.g. mean (SD) ratio for $n = 100$ and regular design was 1.01 (0.1) for all CV strategies). Nevertheless, for clustered and non-uniform designs, while LOO CV appeared to underestimate the error of the maps predicted by the benchmark models (e.g. mean (SD) ratio 0.91 (0.07) for $n = 150$ and clust2 design), it was correctly addressed when using both $\text{sbLOO}_{\text{range}}$ (mean

ratio (SD) 1.01 (0.08)) and $\text{sbLOO}_{\text{ndm}}$ (mean ratio (SD) 1.01 (0.08)).

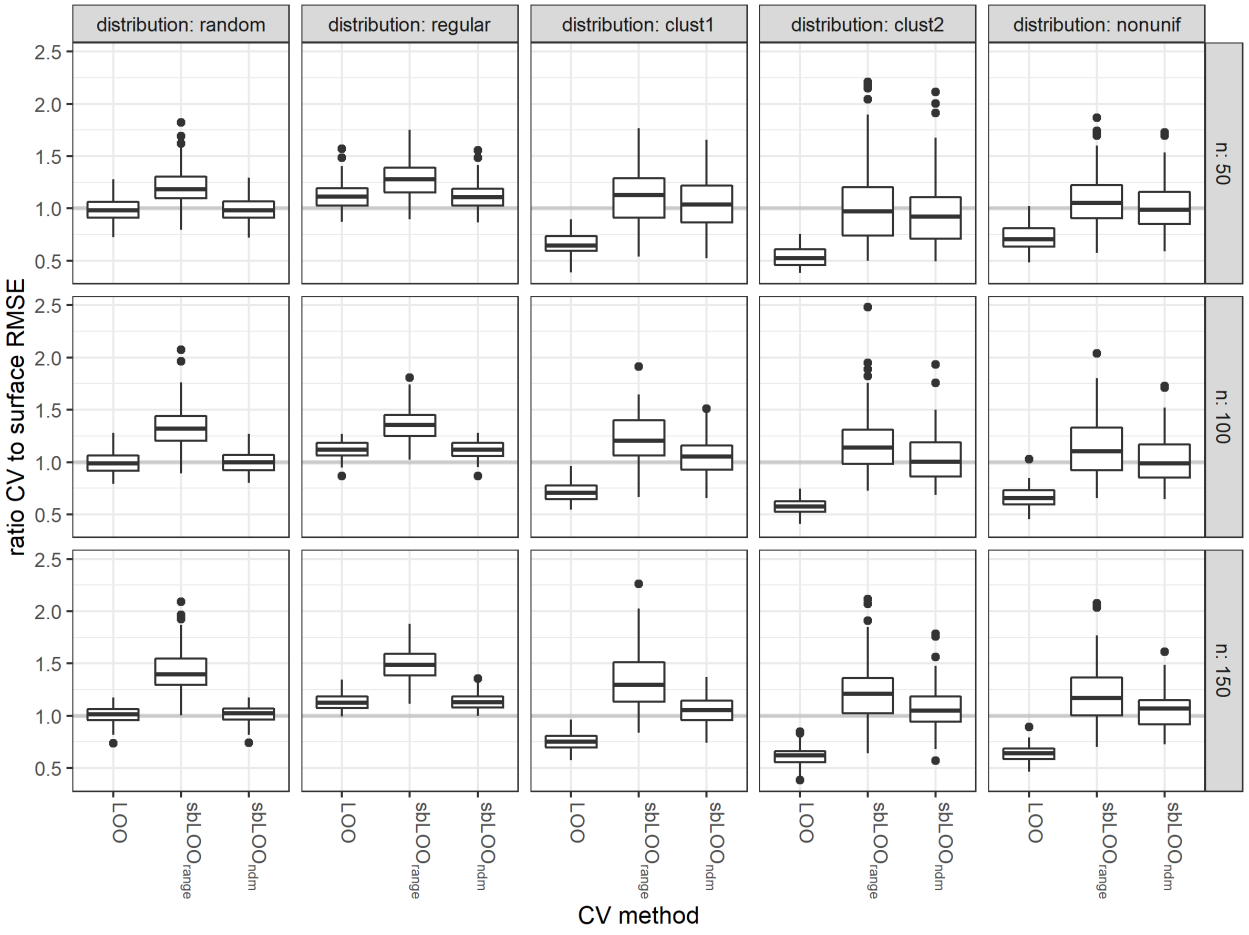


Figure 4.5: Ratio of CV to surface RMSE of benchmark models by sampling distribution and size for high spatial autocorrelation range (40%). Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

For longer ranges (40%), results for the benchmark model and the RMSE statistic were strongly influenced by both the distribution of the sampling points and the CV method used (Figure 4.5). Regardless of the sample size, while LOO correctly estimated the RMSE for random designs (e.g. $n = 100$, mean (SD) ratio 1 (0.1)), it overestimated it for regular designs (e.g. $n = 100$, mean (SD) ratio 1.12 (0.08)) and underestimated it for clustered and non-uniform samples (e.g. $n = 100$, clust1, mean (SD) ratio 0.71 (0.1)). On the other hand, $\text{sbLOO}_{\text{range}}$ resulted in overestimation of the RMSE for all sample sizes and distributions which, in some cases, was large (e.g. mean ratio (SD) for the random design, $n = 150$ was 1.43 (0.2)). Finally, $\text{sbLOO}_{\text{ndm}}$ yielded the same estimated values than LOO for random and regular designs while adequately addressing underestimation in the clustered and non-uniform designs: for $n = 100$, mean $\text{sbLOO}_{\text{ndm}}$ (SD) was 1 (0.1) for the random design, 1.12 (0.08) for regular, 1.05 (0.17) for clust1, 1.05 (0.24) for clust2, and 1.03 (0.23) for non-uniform. Generally, variability of RMSE ratios for $\text{sbLOO}_{\text{ndm}}$ was lower than that of $\text{sbLOO}_{\text{range}}$, though still greater than that of LOO.

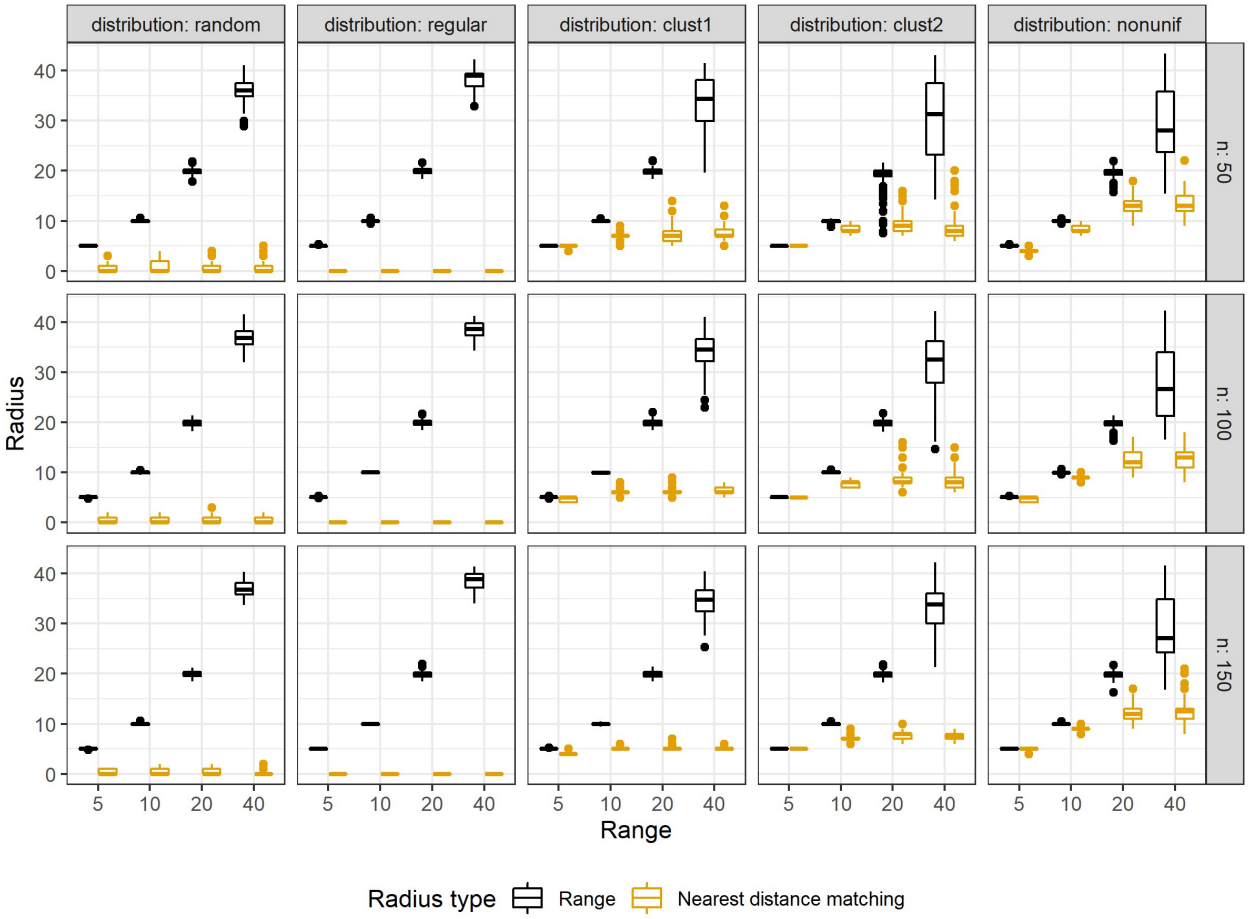


Figure 4.6: Radii for sbLOO strategies by range, and sample size and distribution. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

Examination of the radii used for sbLOO CV indicated that range radii were roughly equal to the landscape range simulation parameter except for range=40, where the condition imposed that all CV iterations must include a 50% of the total sample size reduced them to lower numbers (see respective methods section 3.5.1). On the other hand, NDM yielded radii equal or near 0 for regular and random designs independently of the sample size and range. Regarding clustered and non-uniform designs, those with higher degree of clustering (clust2) generally had NDM greater radii than those with a milder degree (clust1); for all of them larger NDM radii were observed with increasing ranges. All in all, NDM radii were equal or smaller than those purely based on the range. Note that radii included in Figure 4.6 are valid for all models, since none of the two included methods for radius search depended on the modelling strategy.

Results for RMSE ratios and $n = 100$ for all models and ranges (supplementary Figure B.5) indicated that results were the same regardless of the model fitted, and that ranges 10% and 20% showed intermediate patterns of those shown in Figures 4.4 and 4.5 for ranges 5% and 40%, respectively. Results for MAE ratios led to the same conclusions as results for RMSE (sup-

plementary Figure B.6). Differences between CV and surface R^2 for $n = 100$ (supplementary Figure B.7) also went into the same direction; yet, unlike RMSE/MAE, for long ranges and clustered and non-uniform designs $\text{sbLOO}_{\text{ndm}}$ R^2 was similar to $\text{sbLOO}_{\text{range}}$ and both methods still underestimated the actual performance of the model.

Chapter 5

Discussion

5.1 Modelling

The first research question of the Master Thesis was to assess under which spatial autocorrelation and sampling scenarios spatially-explicit ML models were appropriate for spatial interpolation problems. In order to investigate it, we run a simulation study fitting, for different autocorrelation and sampling scenarios, a non-spatial RF model with the simulated predictors and 3 spatially-explicit models with additional covariates: coordinates, EDF, and RFsp. Performance was assessed using metrics comparing the simulated outcome and predicted surfaces.

We found that, for small ranges, non-spatial RF models were superior to spatially-explicit models. We think this is a natural consequence of the lack of spatial structures in the data; consequently, proxies of the geographic locations that aim to describe them will be noise in the models. Indeed, the worse model in all cases was RFsp, which had the largest number of additional spatial predictors. We did not find any example in the literature directly exploring this case; however, in the study by Rocha et al. (2020), non-spatial statistical and ML models were compared to a simple spatial regression model for a variety of simulated landscapes and sampling designs. Unlike our results, Rocha and colleagues found that, for short landscape autocorrelation ranges, the spatial regression model yielded similar, but not worse, errors than a non-spatial linear model. That said, the methods to account for residual spatial autocorrelation in Rocha et al. (2019) via modelling of spatial random effects were different than ours.

For long ranges, the most appropriate model depended on the sampling design: while for regular and random designs spatially-explicit models were more appropriate than non-spatial models, for clustered and non-uniform designs they were equivalent, if not worse. We think that these contrasting results directly relate to the trade-off described by Roberts et al. (2017): ignoring (residual) spatial structures in spatial prediction interpolation may result in a suboptimal model, but at the same time including non-causal predictors that share the same structure might lead to overfitting. We found that the key of this trade-off is, in fact, the sampling design, which had already been pointed out as a possibility in Hengl et al. (2018); Meyer et al. (2019). In that direction, the study by Rocha et al. (2020) mentioned in the previous paragraph found that for long autocorrelation ranges, the simple spatial regression model was clearly superior

to non-spatial linear and more complex ML models for both random and regular samples. Unfortunately, clustered designs were not included in their study.

Regarding modelling with coordinates and similarly to our results, Meyer et al. (2019) found that both for a leaf area index regression and land use land cover classification with strongly autocorrelated outcomes and predictors, as well as clustered samples, performance of the models with coordinates was similar or worse than non-spatial models when validated using spatial CV methods. Our coordinate results also agree with the findings of Cracknell and Reading (2014) for geological classification mapping, who found that models with environmental predictors and coordinates were superior to models using environmental predictors only when the number of sampling clusters was very large (i.e. equivalent to a random design). Our predicted maps showing artefacts when using geographic predictors agree with the patterns found in the maps of Cracknell and Reading (2014); Li et al. (2011); Meyer et al. (2019).

EDF models were compared to coordinates ML models in the study by Behrens et al. (2018), where two examples on spatial interpolation of soil components were presented. Judging by the maps included in the article, in both cases the autocorrelation of the outcome was strong; and while in one case the distribution of the samples appeared to be slightly regular, in the other it looked non-uniform. Behrens and colleagues found that, for both cases, models including coordinates as predictors performed worse than models with EDF. While this agrees with our results for regular sampling, it contrasts with our results for non-uniform designs, which indicate a similar performance of coordinates vs. EDF.

Regarding RFsp, Sekulić et al. (2020) compared RFsp with a RF with coordinates in the predictor set (among other models) for spatio-temporal interpolation of two meteorological outcomes exhibiting large autocorrelation, and sampling designs between random and regular. In agreement with our results, the authors found that RFsp yielded better or equivalent results than RF with coordinates for those sampling designs.

Finally and more generally, our modelling results can be related to the spatial sampling literature. First, Wadoux et al. (2019) recommended using a sampling design spread in the feature space rather than in the geographic space for predictive mapping using RF. However, when coordinates and distance fields are added to the feature space, the optimal design would be a mixture of both. Looking at the geostatistics literature, we see that Heuvelink et al. (2006) argue that, for Regression Kriging, the optimal design should balance the feature and the geographical space, and that the latter should be prioritised if the stochastic component (i.e. the residual spatial structure) is strongly autocorrelated. The findings of these two studies agree with our results, as spatially-explicit ML models have been found to be more effective with sampling designs spread in space in large autocorrelation range settings.

5.2 Validation

The second research question was to find out under which spatial autocorrelation and sampling scenarios spatially-explicit validation strategies were able to better estimate the actual performance of the interpolated maps. To do so, we validated each of the ML models fitted in the sandbox using LOO, $\text{sbLOO}_{\text{range}}$ and $\text{sbLOO}_{\text{ndm}}$ CV and compared them to their surface counterparts.

Our results indicated that the main driver of the patterns found for CV results was the sampling design. While LOO correctly estimated the interpolation error for random designs, it resulted in error overestimation in regular and underestimation in clustered sampling designs. The sample size did not seem to play a role in this process, while larger autocorrelation ranges only increased the size of these effects: over/underestimation was unnoticeable for short ranges but was important for long ranges. The effects of spatial sampling and autocorrelation range on CV for spatial interpolation were explored in Rocha et al. (2020). Briefly, the authors compared the performance (RMSE) of spatial prediction models for a variety of simulated landscapes and designs using random 10-fold CV, a resampling technique comparable to LOO CV, and an independent set of test points. They found that RMSEs estimated using 10-fold CV and independent samples were similar for small ranges but, as the range increased, the systematic (regular) design had the greatest 10-fold CV error while having the smallest test sample error, whereas the opposite was true for close pair and in-fill designs (more similar to clustered designs). Even though the authors did not directly measure whether these methods correctly estimated the true error of the simulated outcomes, the patterns found in relation to the sampling distribution and range agree with our results.

Results for $\text{sbLOO}_{\text{range}}$, i.e. sbLOO with radius equal to the median autocorrelation range of the covariates, indicated that for long ranges, $\text{sbLOO}_{\text{range}}$ overestimated the interpolation error in all designs. Namely for clustered designs, although it did correct the underestimation of the error under LOO, estimates of the error were now inflated. Roberts et al. (2017) performed a simulation study examining random and sbLOO CV with different radii, and compared their ability to estimate the true RMSE of new independent samples. Roberts and colleagues concluded that random CV errors yielded underestimated RMSEs whereas sbLOO with radius equal to the residual range resulted in good error estimates. Even though this does not agree with our results, it must be noted that, in the study by Roberts and colleagues, CV results were compared to the prediction error of new independent landscape locations (i.e. geographic extrapolation), which does not apply to spatial prediction interpolation in the geographic space, where both dependent and independent prediction locations are expected.

Our proposed method $\text{sbLOO}_{\text{ndm}}$ seemed to correctly address underestimation of the error of LOO under clustered and non-uniform sampling designs. For random and regular designs, error estimates were similar or equal to those of LOO. Our NDM algorithm returned sbLOO radii that were smaller than the landscape range, thus avoiding unnecessary extrapolation in the predictor space (Roberts et al., 2017) and keeping the largest sample size possible in each resampling iteration. Even though the potential relevance of the sampling design for spatial CV has been mentioned in previous studies (Hengl et al., 2018; Meyer et al., 2019), this is the first attempt to incorporate them into a method. In our study, $\text{sbLOO}_{\text{ndm}}$ performed better than $\text{sbLOO}_{\text{range}}$ for spatial interpolation model validation. We think this is because the existing methods and guidelines to choose block size/radii for spatial CV methods were designed to estimate extrapolation error. Indeed, most of the literature concerned with these issues come from ecology (e.g. Telford and Birks (2009); Valavi et al. (2019); Wenger and Olden (2012)), where model transferability is of utmost interest. In that context, setting radii that guarantee independence between training and held-out data makes sense, as the relationships between the training and prediction data should be emulated during error assessment (Roberts et al., 2017). However, we argued that in instances where spatial interpolation in the geographical space is needed and hence independence between the training and prediction locations does not always hold, the distribution of the samples must also be considered in the CV. Still, further testing of our proposed method using real-world datasets is required to confirm its potential.

5.3 Recommendations

As a summary of our findings and in order to offer guidance to spatial prediction practitioners, we designed a graphical summary of our results in form of analysis steps and decision trees to help to decide whether spatially-explicit models are recommended for spatial interpolation problems depending on the landscape and sampling data available (Figure 5.1).

The first step consists in performing a thorough spatial exploratory analysis of outcome, predictors, and spatial locations of the samples, in addition to usual predictive modelling exploratory analyses. This is possibly the most important step, since decisions on modelling and validation will depend on it. A practical introduction to these methods can be found in Bivand et al. (2008). The first element to be explored is the landscape autocorrelation, which we suggest to encompass both the outcome measured at the sample locations, as well as the continuous predictors. Although by simply mapping the data we can have a first impression of the smoothness/roughness of the fields, we recommend to estimate empirical semivariograms to have an indication of the degree of autocorrelation in the data. An alternative possibility is to explore autocorrelation via Moran's I correlogram. The estimated range can be expressed as a % of the study area length in one direction, or any other relevant distance benchmark.

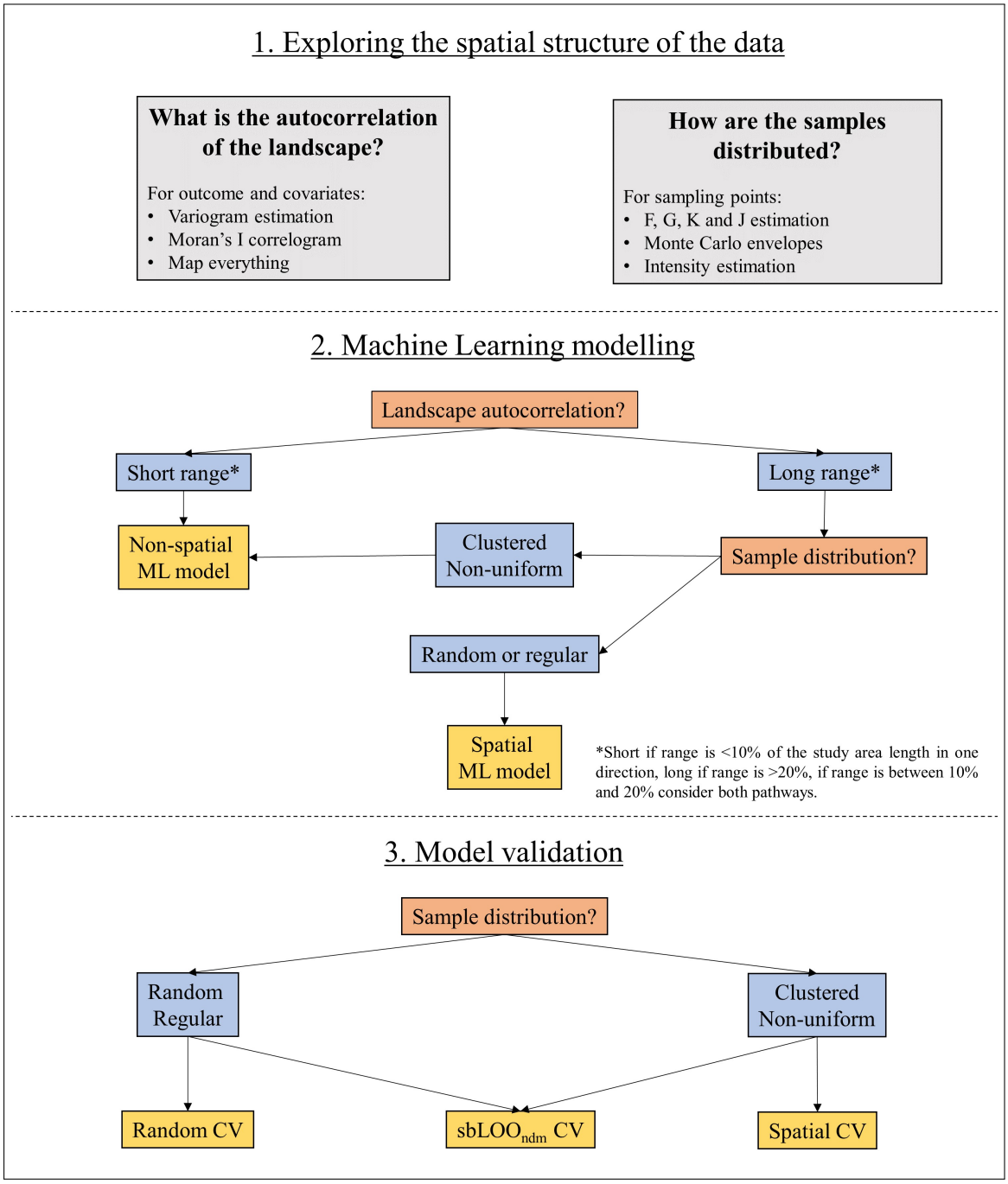


Figure 5.1: Suggested steps to decide on whether to use of spatially-explicit ML and validation methods in spatial interpolation problems.

Secondly, the distribution of the samples needs to be explored. We can assess whether our samples follow a random, regular, or clustered pattern via estimation of the F, G, K, and J functions and their Monte Carlo envelopes, from which we can get insights regarding possible departures from CSR. If a visual inspection or knowledge of the sampling data seems to indicate that the distribution may be non-uniform, a non-parametric intensity estimation might help determining if it is indeed the case. If the locations of the samples are suspected to be correlated

to one existing covariate, the association can be examined via parametric intensity estimation.

Once information about the landscape autocorrelation and the distribution of the samples has been gathered, the analyst can follow the decision trees in steps 2 and 3 to check whether a spatial model/validation is likely to be the best choice for their spatial interpolation problem. Nonetheless, we still recommend to fit a range of possible models including both non-spatial and spatial models, and validate them using both random and spatial CV methods. Large differences between random and spatial CV metrics will indicate that the influence of the sample distribution and the landscape autocorrelation is important and needs to be considered. Likewise, fitting both spatial and non-spatial models and comparing them with the appropriate CV approach may help to confirm that a given model is the most appropriate for a specific application, as differences between models in our simulation analyses were sometimes small.

5.4 Strengths and limitations

Our study has many strengths. Firstly, it is the first to do a comprehensive evaluation of recently proposed spatially-explicit ML models and validation strategies for spatial interpolation problems depending on the landscape autocorrelation and the sample size and distribution. Secondly, it proposes a spatial prediction sandbox framework which grants flexibility for changing many parameters and adding extensions to test other hypotheses. Finally, it proposes a new algorithm for sbLOO CV radius estimation after having identified that the current proposals based solely on the autocorrelation range are not adequate for spatial interpolation problems.

Some limitations must also be acknowledged. First, we could not fully parallelize the sandbox code due to random seed management problems in the HPC cluster, nor could we use more recent and efficient libraries for RF model fitting, namely `ranger`, due to failure in hyper-threading management in the cluster. These two issues made our code not to be as efficient as possible and made the successive sandbox runs to be rather long. Secondly, we did not perform hyperparameter tuning of the RF models but rather took default values for the parameters to shorten computation times. Even though RF has been acknowledged to be quite insensitive to (spatial) hyperparameter tuning (Schratz et al., 2019), better performances when doing so could still be expected (Huang and Boutros, 2016). Thirdly, our estimation of the range parameter was limited to the median range of the covariate fields, which may subject to limitations when using real world data. We chose this method rather than approaches based on the samples' locations to bypass limitations in automated (hence without visual inspection) outcome/residual semivariogram fitting with a limited set of clustered (missing large lags, see Reilly and Gelman 2007) and regular (when the range is smaller than the distance between points, see Müller and Zimmerman 1999) sampling points.

5.5 Potential extensions

Our sandbox simulation framework would allow to extend analyses to a broader range of cases not considered in this Master Thesis. A selection of those are:

- We decided to focus on the regression case, but our analyses could be extended to categorical outcomes.
- Our approach was focused on the purely spatial case. We could however extend the spatial prediction sandbox to the spatio-temporal case.
- Our results focus on spatial interpolation; however, we could also analyse performance of the methods when doing geographic extrapolation.
- Our random field simulation assumed stationarity and isotropy, two assumptions that we could relax.
- We explored the effect of landscape autocorrelation via different range parameters, but we could also extend this idea to different partial sills and nuggets.
- Our approach ignores the effect of spatial heterogeneity, i.e. the equation to generate the outcome remains constant in all the study area. We could apply a spatially-varying generating function.
- Other equations to produce the outcome could be tried in order to see the effect of different signal-to-noise ratios and number of predictors on the results.
- Our proposed NDM method is designed for sbLOO and cannot be used when performing spatial block CV. A version of the algorithm for spatial block CV could be developed.
- Our results were exclusively based on RF models, but we could try alternative ML models as well.
- We did not explore the importance of geographic predictors in spatially-explicit ML models, which could be done by inspecting variable importance statistics of RF models.
- We could explore whether available variable selection methods for spatial prediction (Meyer et al., 2019) would effectively detect in which cases additional geographic predictors in ML models are indeed helpful.

5.6 Conclusions

In this Master Thesis, we set out to investigate under which landscape autocorrelation and sampling scenarios spatially-explicit ML modeling and validation methods were an optimal choice for spatial interpolation problems. We discovered that spatially-explicit ML modelling is only appropriate for long landscape autocorrelation ranges and random and regular sample designs, while spatial CV methods are only recommended for clustered and non-uniform samples. Furthermore, we found out that state-of-the-art methods for spatial CV based only on the

landscape range are not suited for spatial interpolation, and suggested an alternative method based on both the autocorrelation range and the sample distribution.

These results have a large impact on the spatial interpolation field, since it is the first study evaluating a broad range of both standard and spatial ML modelling and CV techniques, and elucidates in what cases one may be preferred over another. Furthermore, our new NDM method for spatial interpolation CV has proved to be effective in our simulation analyses and, though it still needs to undergo further testing, showed promising results. Finally, we provide recommendations on how to decide whether spatially-explicit methods are appropriate based on a spatial exploratory analysis using spatial statistics tools. Our findings benefit all research fields and applications dealing with predictive mapping, ranging from geology, meteorology, forestry, and ecology.

References

Aldrich, C. (2020). Process variable importance analysis by use of random forests in a shapley regression framework. *Minerals*, 10(5):420.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial point patterns: methodology and applications with R*. CRC press.

Baddeley, A. and Turner, R. (2005). spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42.

Beelen, R., Hoek, G., Vienneau, D., Eeftens, M., Dimakopoulou, K., Pedeli, X., Tsai, M.-Y., Künzli, N., Schikowski, T., Marcon, A., Eriksen, K. T., Raaschou-Nielsen, O., Stephanou, E., Patelarou, E., Lanki, T., Yli-Tuomi, T., Declercq, C., Falq, G., Stempfelet, M., Birk, M., Cyrys, J., von Klot, S., Nádor, G., Varró, M. J., Dėdelė, A., Gražulevičienė, R., Mölter, A., Lindley, S., Madsen, C., Cesaroni, G., Ranzi, A., Badaloni, C., Hoffmann, B., Nonnemacher, M., Krämer, U., Kuhlbusch, T., Cirach, M., de Nazelle, A., Nieuwenhuijsen, M., Bellander, T., Korek, M., Olsson, D., Strömgren, M., Dons, E., Jerrett, M., Fischer, P., Wang, M., Brunekreef, B., and de Hoogh, K. (2013). Development of no2 and nox land use regression models for estimating air pollution exposure in 36 study areas in europe – the escape project. *Atmospheric Environment*, 72:10 – 23.

Behrens, T., Schmidt, K., Viscarra Rossel, R. A., Gries, P., Scholten, T., and MacMillan, R. A. (2018). Spatial modelling with euclidean distance fields and machine learning. *European journal of soil science*, 69(5):757–770.

Bivand, R. S., Pebesma, E. J., and Gómez-Rubio, V. (2008). *Applied spatial data analysis with R*, volume 747248717. Springer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Čeh, M., Kilibarda, M., Lisec, A., and Bajat, B. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. *ISPRS international journal of geo-information*, 7(5):168.

Corporation, M. and Weston, S. (2019). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.15.

Cracknell, M. J. and Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, 63:22 – 33.

Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

de Hoogh, K., Chen, J., Gulliver, J., Hoffmann, B., Hertel, O., Ketzel, M., Bauwelinck, M., van Donkelaar, A., Hvidtfeldt, U. A., Katsouyanni, K., Klompmaker, J., Martin, R. V., Samoli, E., Schwartz, P. E., Stafoggia, M., Bellander, T., Strak, M., Wolf, K., Vienneau, D., Brunekreef, B., and Hoek, G. (2018). Spatial pm2.5, no2, o3 and bc models for western europe – evaluation of spatiotemporal stability. *Environment International*, 120:81 – 92.

Diggle, P. J., Menezes, R., and Su, T.-l. (2010). Geostatistical inference under preferential sam-

pling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 59(2):191–232.

Fick, S. E. and Hijmans, R. J. (2017). Worldclim 2: new 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37(12):4302–4315.

Folberth, C., Skalskỳ, R., Moltchanova, E., Balkovič, J., Azevedo, L. B., Obersteiner, M., and Van Der Velde, M. (2016). Uncertainty in soil data can outweigh climate impact signals in global crop yield simulations. *Nature communications*, 7(1):1–13.

Fourcade, Y., Besnard, A. G., and Secondi, J. (2018). Paintings predict the distribution of species, or the challenge of selecting environmental predictors and evaluation statistics. *Global Ecology and Biogeography*, 27(2):245–256.

Gebbers, R. and de Bruin, S. (2010). *Application of Geostatistical Simulation in Precision Agriculture*, pages 269–303. Springer Netherlands, Dordrecht.

Georganos, S., Grippa, T., Gadiaga, A. N., Linard, C., Lennert, M., Vanhuysse, S., Mboga, N., Wolff, E., and Kalogirou, S. (2019). Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling. *Geocarto International*, 0(0):1–16.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hengl, T., Heuvelink, G. B., and Rossiter, D. G. (2007). About regression-kriging: From equations to case studies. *Computers & Geosciences*, 33(10):1301 – 1315. Spatial Analysis.

Hengl, T., Mendes de Jesus, J., Heuvelink, G. B., Ruiperez Gonzalez, M., Kilibarda, M., Blagotić, A., Shangguan, W., Wright, M. N., Geng, X., and Bauer-Marschallinger, B. (2017). Soilgrids250m: Global gridded soil information based on machine learning. *PLoS one*, 12(2):e0169748.

Hengl, T., Minasny, B., and Gould, M. (2009). A geostatistical analysis of geostatistics. *Scientometrics*, 80(2):491–514.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., and Gräler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, 6:e5518.

Heuvelink, G. B., Brus, D. J., and de Gruijter, J. J. (2006). Chapter 11 optimization of sample configurations for digital mapping of soil properties with universal kriging. In Lagacherie, P., McBratney, A., and Voltz, M., editors, *Digital Soil Mapping*, volume 31 of *Developments in Soil Science*, pages 137 – 151. Elsevier.

Hiemstra, P., Pebesma, E., Twenhofel, C., and Heuvelink, G. (2008). Real-time automatic interpolation of ambient gamma dose rates from the dutch radioactivity monitoring network. *Computers & Geosciences*. DOI: http://dx.doi.org/10.1016/j.cageo.2008.10.011.

Hijmans, R. J. (2019). *raster: Geographic Data Analysis and Modeling*. R package version 2.9-23.

Huang, B. F. and Boutros, P. C. (2016). The parameter sensitivity of random forests. *BMC bioinformatics*, 17(1):331.

Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*, volume 26. Springer.

Lary, D. J., Alavi, A. H., Gandomi, A. H., and Walker, A. L. (2016). Machine learning in

geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3 – 10. Special Issue: Progress of Machine Learning in Geosciences.

Le Rest, K., Pinaud, D., Monestiez, P., Chadoeuf, J., and Bretagnolle, V. (2014). Spatial leave-one-out cross-validation for variable selection in the presence of spatial autocorrelation. *Global ecology and biogeography*, 23(7):811–820.

Li, J. and Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53:173–189.

Li, J., Heap, A. D., Potter, A., and Daniell, J. J. (2011). Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling & Software*, 26(12):1647 – 1659.

Li, T., Shen, H., Yuan, Q., Zhang, X., and Zhang, L. (2017). Estimating ground-level pm2.5 by fusing satellite and station observations: A geo-intelligent deep learning approach. *Geophysical Research Letters*, 44(23):11,985–11,993.

Liao, Y., Li, D., and Zhang, N. (2018). Comparison of interpolation models for estimating heavy metals in soils under various spatial characteristics and sampling methods. *Transactions in GIS*, 22(2):409–434.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. (2005). *Geographic information systems and science*. John Wiley & Sons.

Meyer, H., Katurji, M., Appelhans, T., Müller, M. U., Nauss, T., Roudier, P., and Zawar-Reza, P. (2016). Mapping daily air temperature for antarctica based on modis lst. *Remote Sensing*, 8(9):732.

Meyer, H. and Pebesma, E. (2020). Predicting into unknown space? estimating the area of applicability of spatial prediction models. *arXiv preprint arXiv:2005.07939*.

Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., and Nauss, T. (2018). Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. *Environmental Modelling & Software*, 101:1 – 9.

Meyer, H., Reudenbach, C., Wöllauer, S., and Nauss, T. (2019). Importance of spatial predictor variable selection in machine learning applications – moving from data reproduction to spatial prediction. *Ecological Modelling*, 411:108815.

Misiuk, B., Diesing, M., Aitken, A., Brown, C. J., Edinger, E. N., and Bell, T. (2019). A spatially explicit comparison of quantitative and categorical modelling approaches for mapping seabed sediments using random forest. *Geosciences*, 9(6):254.

Molinaro, A. M., Simon, R., and Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307.

Müller, W. G. and Zimmerman, D. L. (1999). Optimal designs for variogram estimation. *Environmetrics: The official journal of the International Environmetrics Society*, 10(1):23–37.

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1):439–446.

Pebesma, E. J. (2004). Multivariable geostatistics in s: the gstat package. *Computers & Geosciences*, 30(7):683 – 691.

Phiri, D. and Morgenroth, J. (2017). Developments in landsat land cover classification methods: A review. *Remote Sensing*, 9(9):967.

Ploton, P., Mortier, F., Réjou-Méchain, M., Barbier, N., Picard, N., Rossi, V., Dormann, C., Cornu, G., Viennois, G., and Bayol, N. (2020). Spatial validation reveals poor predictive performance of large-scale ecological mapping models. *Nature communications*, 11(1):1–11.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raaschou-Nielsen, O., Andersen, Z. J., Beelen, R., Samoli, E., Stafoggia, M., Weinmayr, G., Hoffmann, B., Fischer, P., Nieuwenhuijsen, M. J., and Brunekreef, B. (2013). Air pollution and lung cancer incidence in 17 european cohorts: prospective analyses from the european study of cohorts for air pollution effects (escape). *The lancet oncology*, 14(9):813–822.

Reilly, C. and Gelman, A. (2007). Weighted classical variogram estimation for data with clustering. *Technometrics*, 49(2):184–194.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F., and Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929.

Rocha, A. D., Groen, T. A., and Skidmore, A. K. (2019). Spatially-explicit modelling with support of hyperspectral data can improve prediction of plant traits. *Remote Sensing of Environment*, 231:111200.

Rocha, A. D., Groen, T. A., Skidmore, A. K., Darvishzadeh, R., and Willemen, L. (2018). Machine learning using hyperspectral data inaccurately predicts plant traits under spatial dependency. *Remote Sensing*, 10(8).

Rocha, A. D., Groen, T. A., Skidmore, A. K., and Willemen, L. (2020). Role of sampling design when predicting spatially dependent ecological data with remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–12.

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., and Brenning, A. (2019). Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, 406:109–120.

Sekulić, A., Kilibarda, M., Heuvelink, G., Nikolić, M., and Bajat, B. (2020). Random forest spatial interpolation. *Remote Sensing*, 12(10):1687.

Telford, R. and Birks, H. (2009). Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews*, 28(13):1309 – 1316.

Trachsel, M. and Telford, R. (2016). Technical note: Estimating unbiased transfer-function performances in spatially structured environments, clim. past, 12, 1215–1223.

Valavi, R., Elith, J., Lahoz-Monfort, J. J., and Guillera-Arroita, G. (2019). blockcv: An r package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. *Methods in Ecology and Evolution*, 10(2):225–232.

Van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical applications in genetics and molecular biology*, 6(1).

Veloz, S. D. (2009). Spatially autocorrelated sampling falsely inflates measures of accuracy for presence-only niche models. *Journal of Biogeography*, 36(12):2290–2299.

Wadoux, A. M.-C., Brus, D. J., and Heuvelink, G. B. (2019). Sampling design optimization for soil mapping with random forest. *Geoderma*, 355:113913.

Wenger, S. J. and Olden, J. D. (2012). Assessing transferability of ecological models: an underappreciated aspect of statistical validation. *Methods in Ecology and Evolution*, 3(2):260–267.

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.

Wylie, B. K., Pastick, N. J., Picotte, J. J., and Deering, C. A. (2019). Geospatial data mining for digital raster mapping. *GIScience & Remote Sensing*, 56(3):406–429.

Xie, Y., Eftelioglu, E., Ali, R. Y., Tang, X., Li, Y., Doshi, R., and Shekhar, S. (2017). Transdisciplinary foundations of geospatial data science. *ISPRS International Journal of Geo-Information*, 6(12):395.

Appendix A

Nearest distance matching

A.1 Background and hypothesis

Spatial buffer Leave-One-Out cross-validation (sbLOO CV) has been suggested to be a good method to provide realistic error estimates for spatial prediction (Telford and Birks, 2009). It consists in a LOO version in which, for each of the points, a model is fitted not only excluding the held-out observation but also observations within a buffer radius s of that point. By excluding the neighbouring points in each LOO iteration, a larger degree of independence between training and held-out data is achieved.

sbLOO requires choosing a parameter, namely the radius s of the buffer. We want to find an s so that sbLOO CV correctly estimates the true surface error for interpolation problems. As explained in section 2.2.3, literature generally suggests choosing s equal to the range of the semivariogram fitted on the model residuals (e.g. Telford and Birks 2009). The reasoning is that we want to have an estimate of the error when we predict for an independent location (i.e. model transferability to new areas/extrapolation). Roberts et al. (2017) warns that in fact s should be at least equal to the residual range because overfitting may have occurred before that, and that the range in the raw outcome should be evaluated instead. Valavi et al. (2019) suggests to define the radius based on the median range of the predictor surfaces.

None of these proposed methods to choose s considers the sampling distribution, but rather they are solely based on the range ϕ to achieve independent held-out samples. However, when we want to interpolate and not to predict to new areas, independence of test samples may not be actually needed, e.g. some of the grid points for which we will predict will be directly at the side of a training point and hence will not be independent from them anyway. Instead, we believe that, for spatial interpolation problems, the appropriate radius s will depend on both the distribution of the samples and the range ϕ. We want that, during our LOO CV process, the distribution of the nearest distances between the held-out and train points resembles as much as possible the distribution of the nearest distances between the grid cells for which we want to predict (here we assume all grid cells in the study area) and the complete set of training data, for those distances in which autocorrelation is present.

For purposes of illustration of the method we propose, we simulate four sets of 100 random, regular, clustered1 (20 parents, 80 offspring, see section 3.3.1), and clustered2 (10 parents, 90 offspring) points in a 100x100 grid.

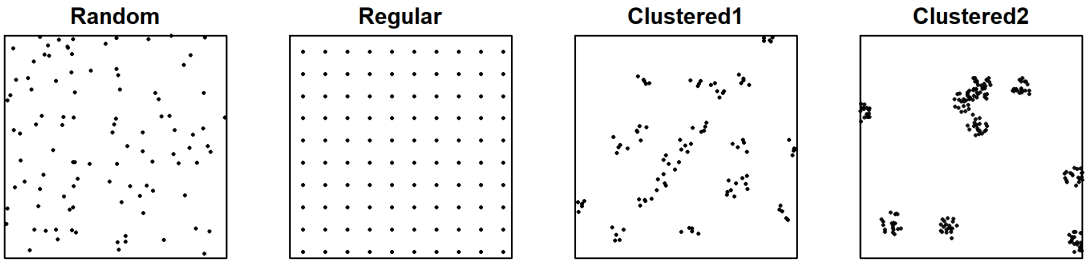


Figure A.1: 100 random, regular, clustered1, and clustered2 simulated points for nearest distance matching illustration.

A.2 Characterising the nearest distance distribution in prediction: The F function

The first step is to characterise the distribution of the distances from each of the grid cells in the study area for which we want to predict to the nearest sampling point. To do so, we can use the F function, known as the *empty space distribution function* in the spatial point process literature (Baddeley et al., 2015). F is the cumulative distribution function measuring, for a fixed location u in the study area, the probability of finding a point $x_i \in \mathbf{x}$ within a distance r of that location, i.e. $F(r) = P[d(u, \mathbf{X}) \leq r]$. When no edge effect is present, F can be estimated by:

$$\hat{F}(r) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}\{d(u_j, \mathbf{x}) \leq r\}$$

where m is the total number of cells and $d(u_j, \mathbf{x}) = \min\{\|u_j - x_i\| : x_i \in \mathbf{x}\}$; i.e., $\hat{F}(r)$ is estimated as the proportion of grid cells that have at least one point within a distance r. Let's see the $\hat{F}$ functions for the simulated points in Figure A.2.

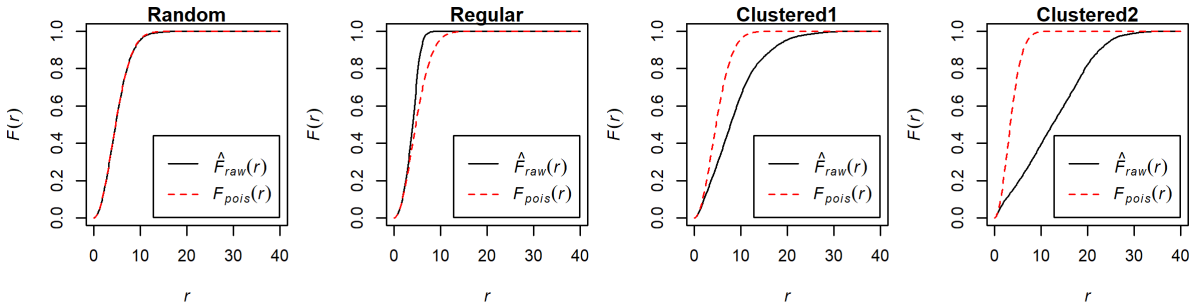


Figure A.2: Theoretical under CRS (red dashed) and estimated (black) F functions for the 4 simulated point sets.

The red dashed line in Figure A.2 represents the theoretical F function under Complete Spatial Randomness (CSR, i.e. events distributed independently at random and homogeneously), so it does not come as a surprise that it coincides with our estimated $\hat{F}$ function (black line) for the simulated random design. Nevertheless, under the other designs, they differ. For the regular design, we see shorter empty space distances that those we would expect under CSR, while for the clustered the opposite is found.

A.3 Characterizing the nearest distance distribution in LOO CV: The G function

The G function is known in the point process literature as the *nearest neighbour distribution function* (Baddeley et al., 2015), and measures the probability of finding, from a location u where one of the points in the sample is present, another point within a distance r of that location; i.e. $G(r) = P[d(u, \mathbf{X} \setminus u) \leq r | \mathbf{X}$ has a point at $u]$. Since in LOO CV we leave out each of the points sequentially, we can characterise the distribution of the nearest distances between the held-out points and the training data during CV using the G function. The empirical $\hat{G}$ distribution function without edge effect can be estimated as follows:

$$\hat{G}(r) = \frac{1}{n} \sum_{i} 1\{d_i \leq r\}$$

where r is a distance, n is the total number of points, and $d_i = \min_{j \neq i} \|x_i - x_j\|$; i.e., $\hat{G}(r)$ is estimated as the proportion of sample points that have another point at a distance equal or lower than r. We compute the G functions of our simulated points in Figure A.3.

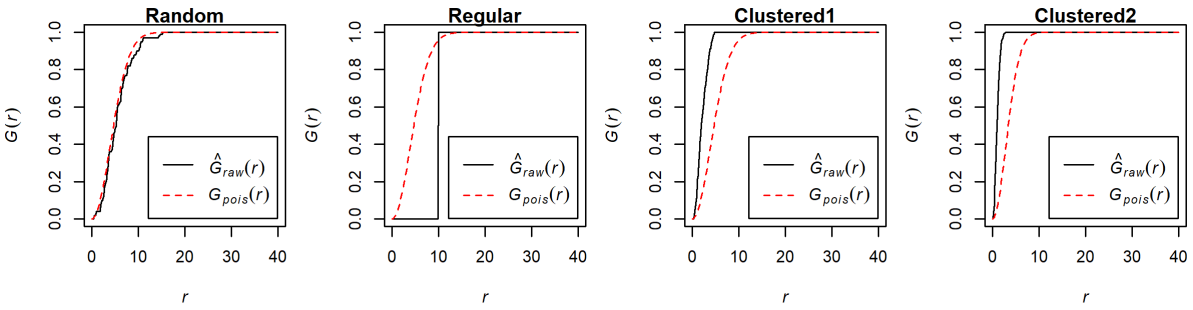


Figure A.3: Theoretical under CRS (red dashed) and estimated (black) G functions for the 4 simulated point sets.

Similarly to the results for the F function, the estimated $\hat{G}$ function (black line) approximates very well the theoretical G line under CSR (red dashed line). For the regular samples, we see that $\hat{G}(r) = 0$ for $r < 10$, and then $\hat{G}(r) = 1$ for $r \geq 10$. This is because in the simulated regular design all (100) points are equally-spaced with a vertical/horizontal spacing of 10 units in a 100x100 grid, and therefore no neighbours can be found at distances lower than 10, but

all points have a neighbour for distances equal to 10 and larger. Finally, we expect a larger number of close neighbours in clustered designs compared to what we would see in a CSR.

A.4 Comparing the G and the F function

Now that we have characterised the nearest distance distribution between training and prediction cells (F function) and LOO CV (G function), we can compare them. Ideally, we would like our estimated $\hat{G}$ function to be as close as possible to the $\hat{F}$ function, so that the nearest distances in the LOO CV reproduce as close as possible those that we will find when predicting the continuous surface. We will do it for our 4 sets of simulated points (Figure A.4).

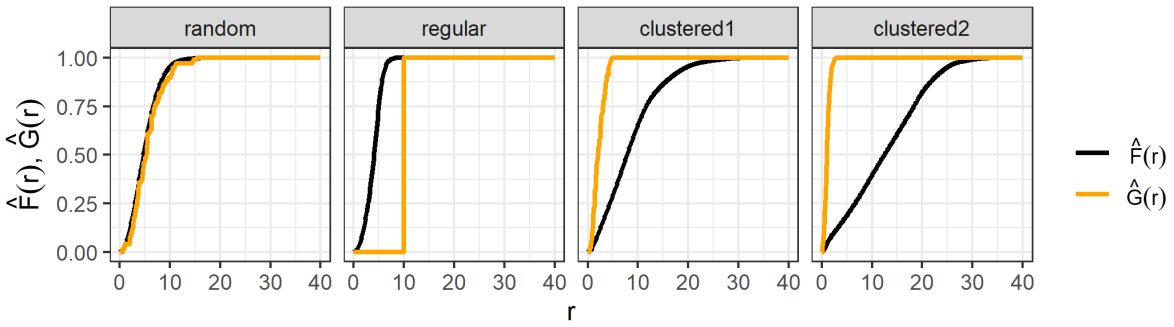


Figure A.4: Comparison of estimated $\hat{F}$ and $\hat{G}$ functions for the 4 simulated point sets.

A visual comparison of the two functions reveals that for random designs the distribution of the nearest distances when performing LOO CV ($\hat{G}$) and when predicting the continuous surface ($\hat{F}$) is very similar, which is not surprising as G and F are equivalent under CSR (Baddeley et al., 2015). On the other hand, we see that the nearest distances in the surface prediction ($\hat{F}$) are shorter than those in the LOO CV ($\hat{G}$) for regular designs. The opposite happens for clustered designs, where nearest distances in the LOO CV are shorter than in prediction.

A.5 Approximating the G function to the F function with sbLOO

Once the differences between the G and the F functions have been analysed, we can try to modify our LOO CV approach so that the $\hat{G}$ function approximates better the $\hat{F}$ function. We can do that by using sbLOO where, for each of the held-out points, all neighbouring points within a radius s are also removed from the data used for training, so that the distances to the nearest points are enlarged. We can adapt the G function to sbLOO by defining a new function $G_b(r, s)$ representing the cumulative distribution function of the nearest neighbours distances with an exclusion buffer of radius s, which can be estimated as:

$$\hat{G}_b(r, s) = \frac{1}{n} \sum_i 1\{d_i^* \leq r\}$$

where $d_i^* = \min_{j \neq i}\{\|x_i - x_j\| : \|x_i - x_j\| > s\}$. That is, $\hat{G}_b(r, s)$ can be estimated as the proportion of points that have, within the set of neighbours at a distance larger than s, a neighbour at a distance equal or smaller than r. We can compute the $\hat{G}_b$ function under a grid of possible radii $\mathbf{s} = \{0, 1, 2, ..., c = \hat{F}^{-1}(0.5)\}$, i.e. a sequence of integers from 0 to the distance c where $\hat{F}(c) = 0.5$. Note that, for practical applications, the appropriate step size for s will depend on the study area dimensions and/or the coordinate reference system in use. We can try this approach with our 4 simulated sets of points (Figure A.5).

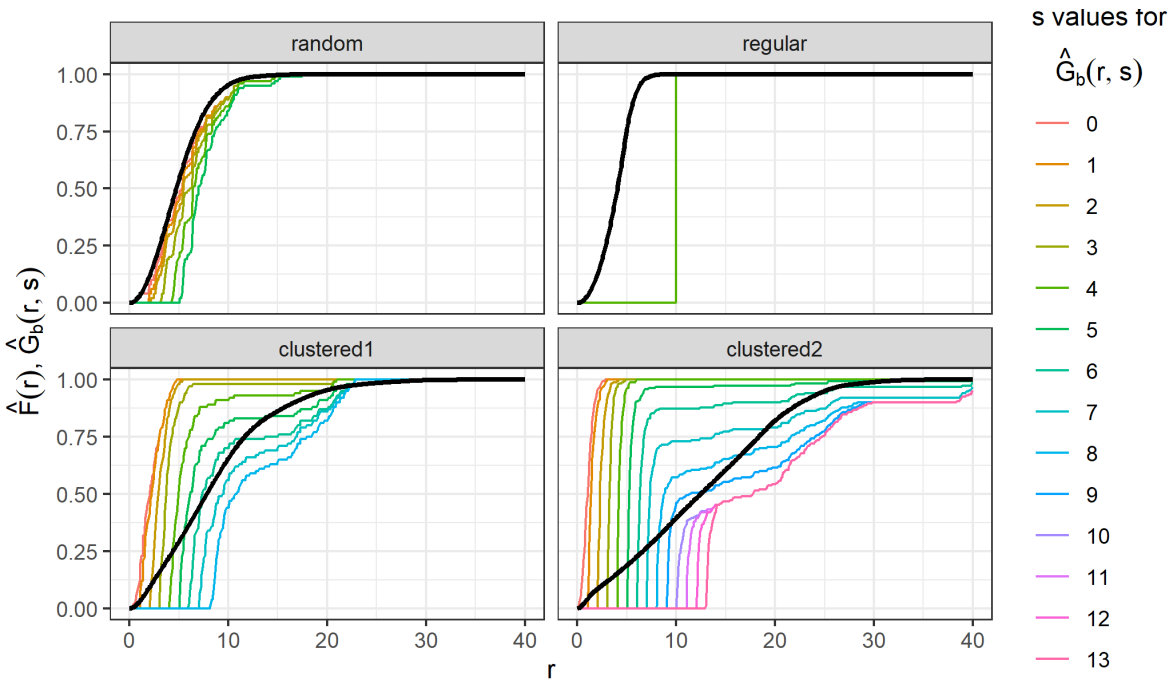


Figure A.5: Comparison of estimated $\hat{F}$ (black bold) and $\hat{G}_b$ functions with different radii (coloured lines) for the 4 simulated point sets.

Some things can be noted from Figure A.5. Firstly, $\hat{G}_b(r, 0) = \hat{G}(r)$, i.e. no buffer is applied. Secondly, $\hat{G}_b(r, s) = 0$ for $r \leq s$ since all points within a radius s are excluded and the condition $(d_i \leq r) \wedge (d_i > s)$ will never be met. Thirdly, the number of $\hat{G}_b$ functions to be estimated will be much larger for clustered designs since $F^{-1}(0.5)$ will also be larger. Finally, all $\hat{G}_b$ functions for the regular case are exactly the same and overlap, as all considered buffer sizes are smaller than the distance between observations and therefore no point in sbLOO is ever excluded.

The final step is to decide which is the radius that will make $\hat{G}_b$ better approximate $\hat{F}$ in each case. We can do that by evaluating the Sum of Squares Error (SSE) between the $\hat{G}_b$ and the $\hat{F}$ function for distances r lower or equal than the (residual/outcome/predictor) estimated autocorrelation range $\hat{\phi}$, i.e. $\mathbf{r} = \{0, 0.1, 0.2, ..., \hat{\phi}\}$ (step size to be optimised in each application). The reasoning for that is that we expect the error to be fairly constant for distances beyond that range, and therefore differences between the two functions beyond it will not be relevant.

With the calculated SSE, we will take the radius $s^* \in \mathbf{s}$ that minimizes that value. If two radii yield the same SEE and it is the minimum, the smallest of them will be taken.

Supposing ϕ to be known and $\phi = 20$, the SSE for the different radii for our 4 simulated samples are included in Table A.1. We find that while s^*, i.e. the radius that minimises the SSE between the G_b and F function for $r \in [0, 20]$ is 0 for the regular and random designs, it is a larger number for the two clustered designs. For the reasons explained before, the SSEs for the regular design are constant.

Radius	random	regular	clustered1	clustered2
0	**0.25**	**51.81**	30.09	70.41
1	0.46	51.81	29.35	67.24
2	0.85	51.81	25.12	59.75
3	2.16	51.81	18.64	53.32
4	5.08	51.81	8.57	46.22
5	8.71	-	**3.38**	34.95
6	-	-	3.57	20.36
7	-	-	6.85	8.68
8	-	-	12.74	**3.88**
9	-	-	-	4.78
10	-	-	-	8.04
11	-	-	-	9.77
12	-	-	-	11.99
13	-	-	-	14.49

Table A.1: SSE between sbLOO's $\hat{G}_b(r, s)$ for different radii and $\hat{F}$ for the simulated sampling designs and autocorrelation range of 20. The radius minimizing the SSE for each simulated sampling design is highlighted.

To conclude, we would like to show the effect of having a different range on s^* (Table A.2). For $\phi = 5$, the optimal radii for the random and clustered design are still 0, yet for the clustered designs they are lower than those for $\phi = 20$. This is because, to calculate the SSE in this case, only $r \in [0, 5]$ have been considered. Also because of that, all $s \geq 5$ yield the same SSE, as $\hat{G}_b(r, s) = 0$ for $r <= s$. Following this result and to avoid unnecessary computation, we can limit the grid of radii to be evaluated to $s_{max} = \min(c, \phi)$ and thus $\mathbf{s} = \{0, 1, 2, ..., s_{max}\}$.

A.6 Summary of the algorithm

We propose a new method named Nearest Distance Matching (NDM) which aims at finding the optimal radius for sbLOO CV for spatial interpolation problems based on the distribution of the samples and the autocorrelation range (Algorithm 1). We expect our algorithm to propose increasing buffer radii for larger degrees of clustering of the samples, buffers close or equal to 0 for random designs, and buffers equal to 0 for regular designs. Radii for clustered designs will be larger for scenarios in which the autocorrelation range is longer, and will tend to 0

Radius	random	regular	clustered1	clustered2
0	**0.10**	**6.15**	13.06	28.32
1	0.26	6.19	12.32	25.16
2	0.51	6.19	8.12	17.66
3	1.37	6.19	3.87	11.23
4	2.94	6.19	**0.78**	4.33
5	3.76	-	1.28	**0.64**
6	-	-	1.28	0.64
7	-	-	1.28	0.64
8	-	-	1.28	0.64
9	-	-	-	0.64
10	-	-	-	0.64
11	-	-	-	0.64
12	-	-	-	0.64
13	-	-	-	0.64

Table A.2: SSE between sbLOO's $\hat{G}_b(r, s)$ for different radii and $\hat{F}$ for the simulated sampling designs and autocorrelation range of 5. The radius minimizing the SSE for each simulated sampling design is highlighted.

for landscapes with no spatial autocorrelation. We expect our algorithm to propose a radius for sbLOO that will solve underestimation of prediction error when using LOO with clustered samples, while yielding the same results as LOO for random and regular designs. Note that step sizes for the s and r grids are optimised for the sandbox, but they should be customised to each particular application according to the size of the study area and/or coordinate reference system.

Data: samples (point data), study area (polygon), estimated range ($\hat{\phi}$)
Result: s^*: radius for sbLOO CV for interpolation ;

initialize;
compute $\hat{F}(r)$ function;
get a distance c such that $\hat{F}(c) = 0.5$;
define maximum radius as $s_{max} = \min(c, \hat{\phi})$;
for $\forall s \in \{0, 1, 2, ..., s_{max}\}$ **do**
  for $\forall r \in \{0, 0.1, 0.2, ..., \hat{\phi}\}$ **do**
    compute $\hat{G}_b(r, s)$ function;
  end
  compute $SSE(s)$ between $\hat{F}(r)$ and $\hat{G}_b(r, s)$ for $r \in \{0, 0.1, 0.2, ..., \hat{\phi}\}$;
end
$s^* = \text{argmin}_s\, SSE(s)$;
finalize;

Algorithm 1: The nearest distance matching algorithm.

Appendix B

Supplementary figures

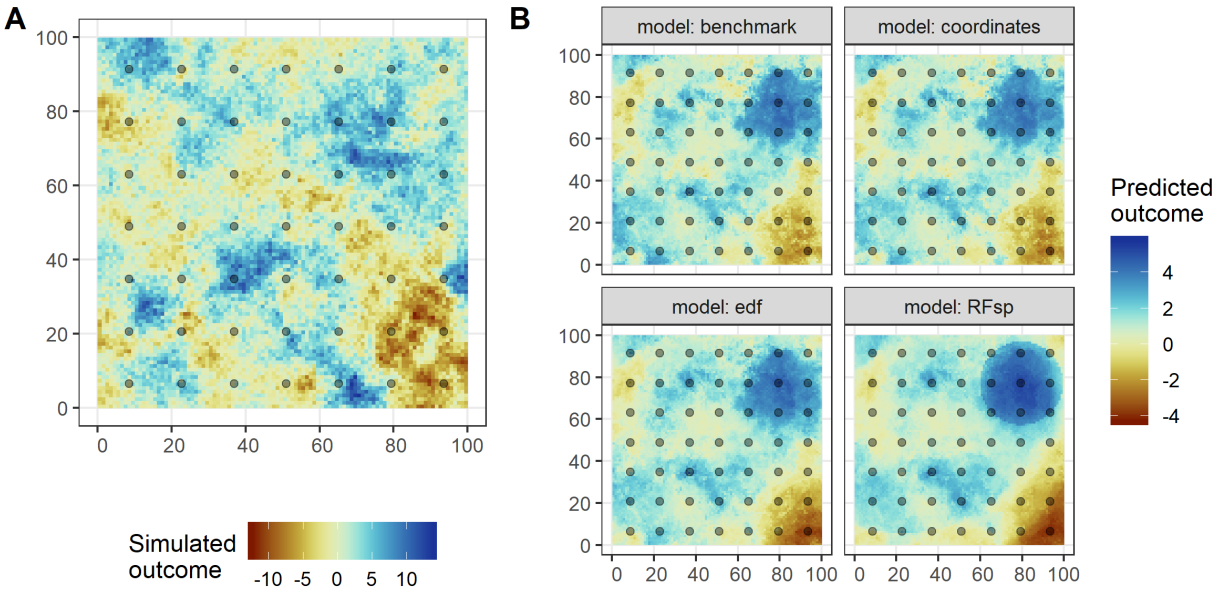


Figure B.1: Example of a simulated outcome surface (A) and four predicted surfaces according to the different included models (B) for $n = 50$, regular distribution, and a landscape with range equal to 40%.

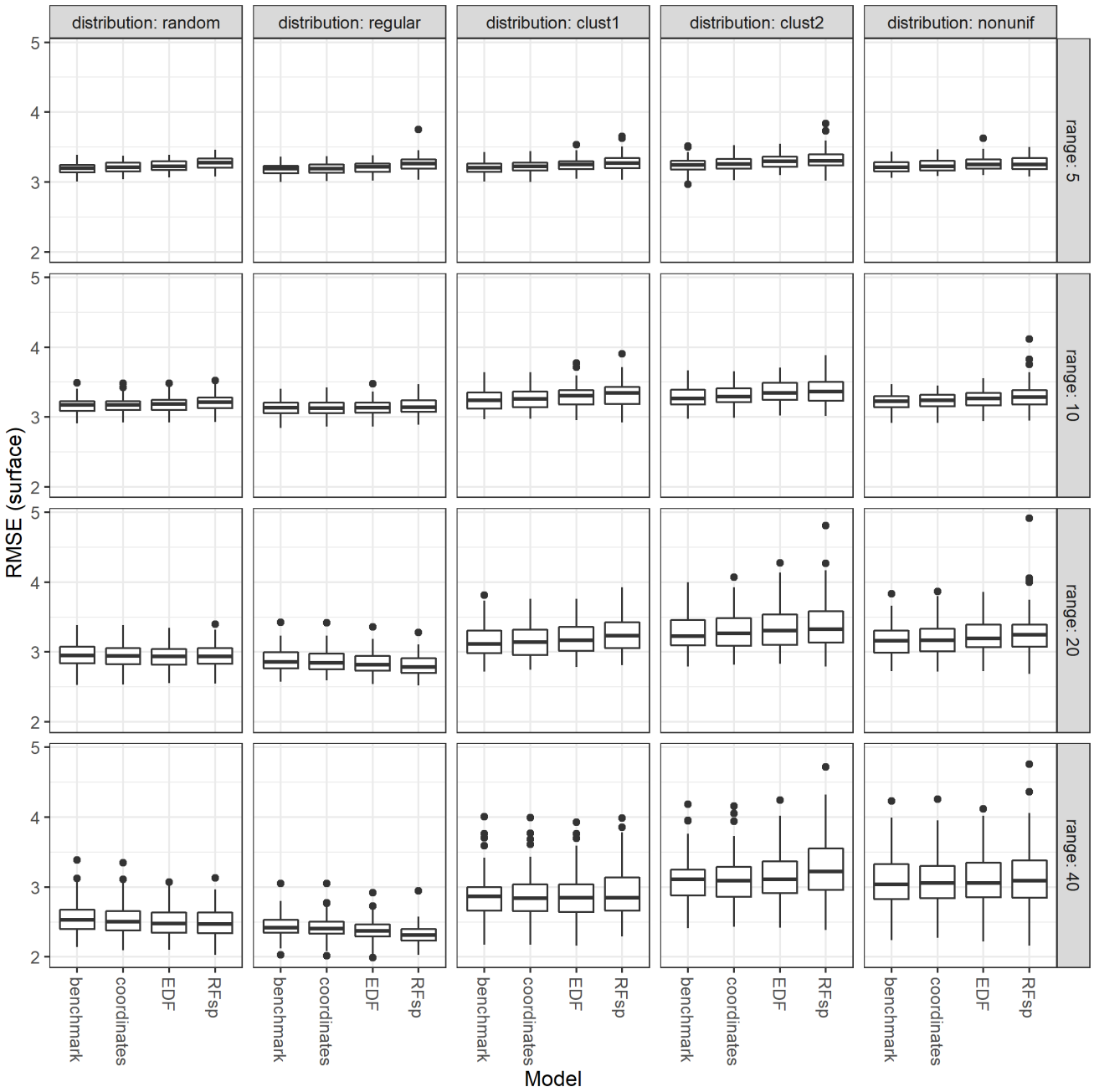


Figure B.2: RMSE (surface) of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

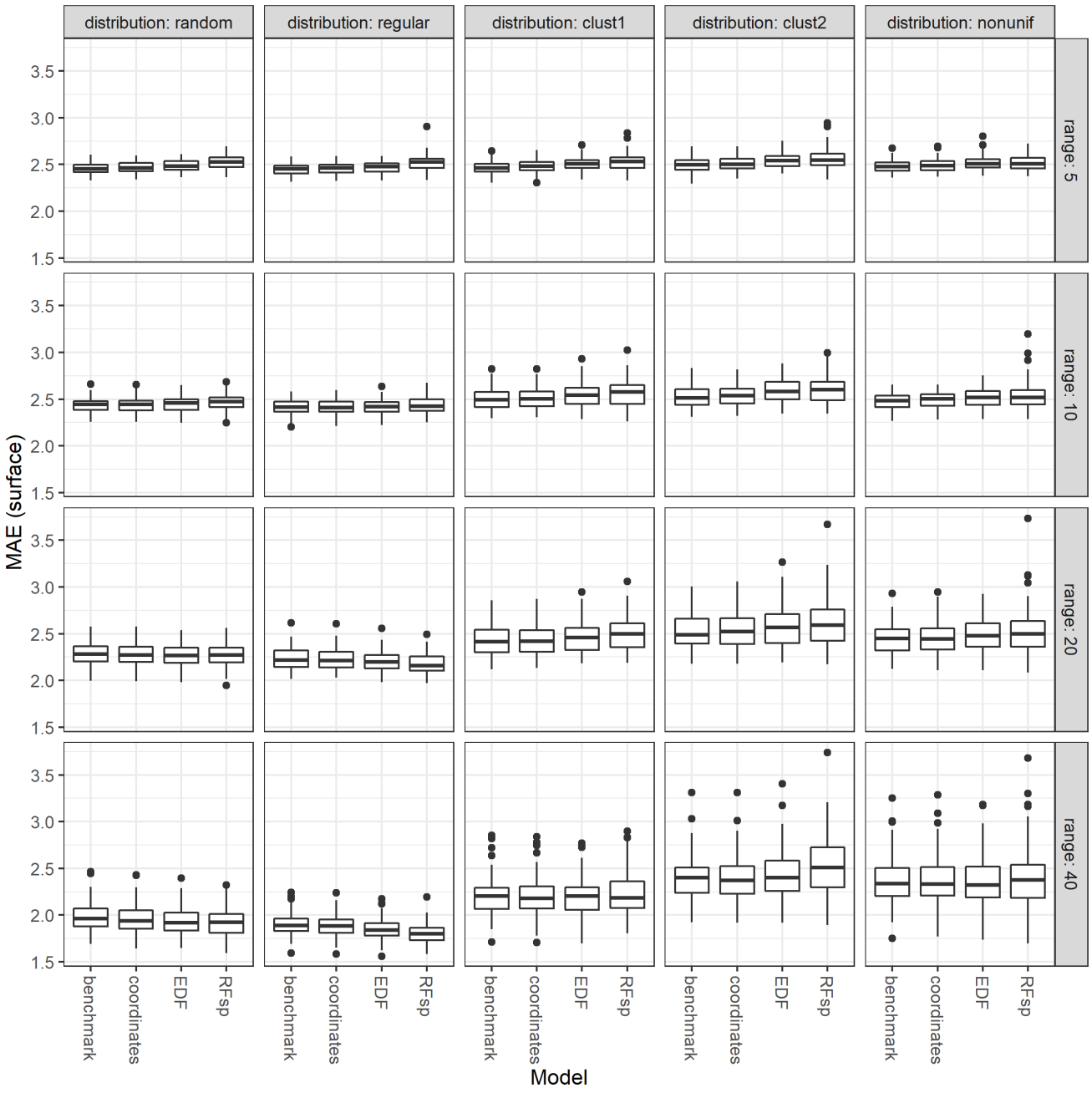


Figure B.3: MAE (surface) of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

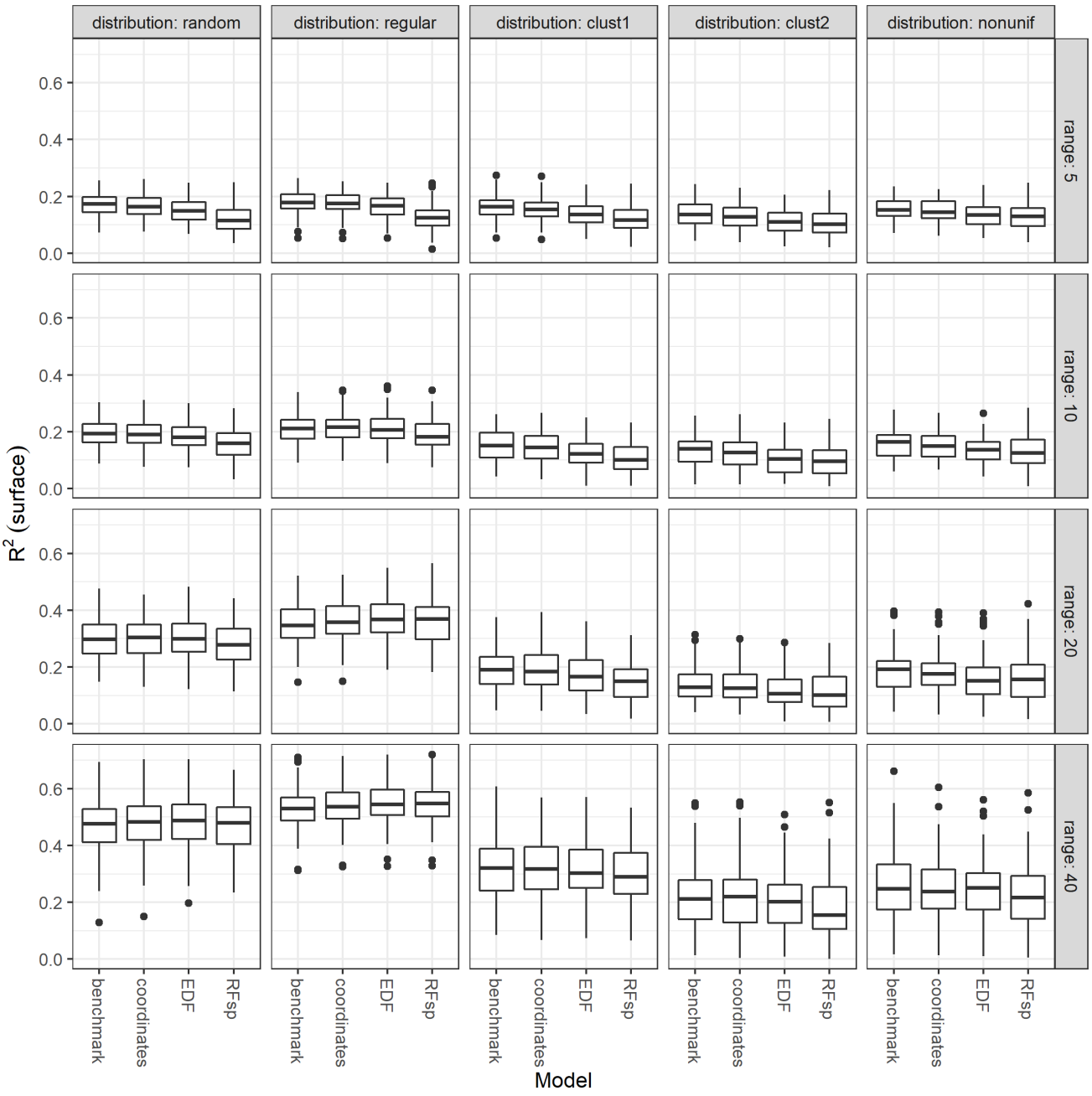


Figure B.4: R^2 (surface) of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

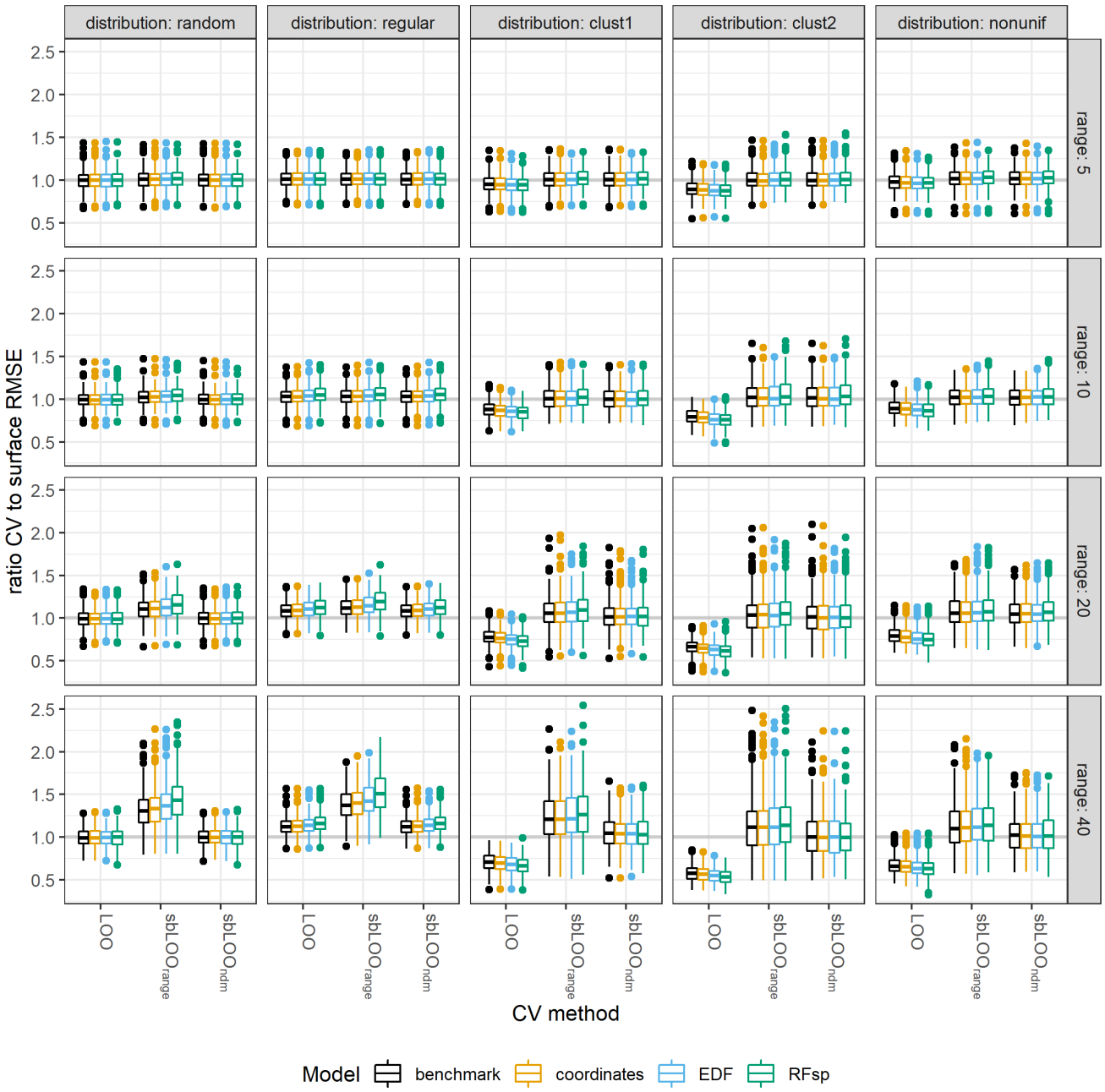


Figure B.5: Ratio of CV to surface RMSE of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

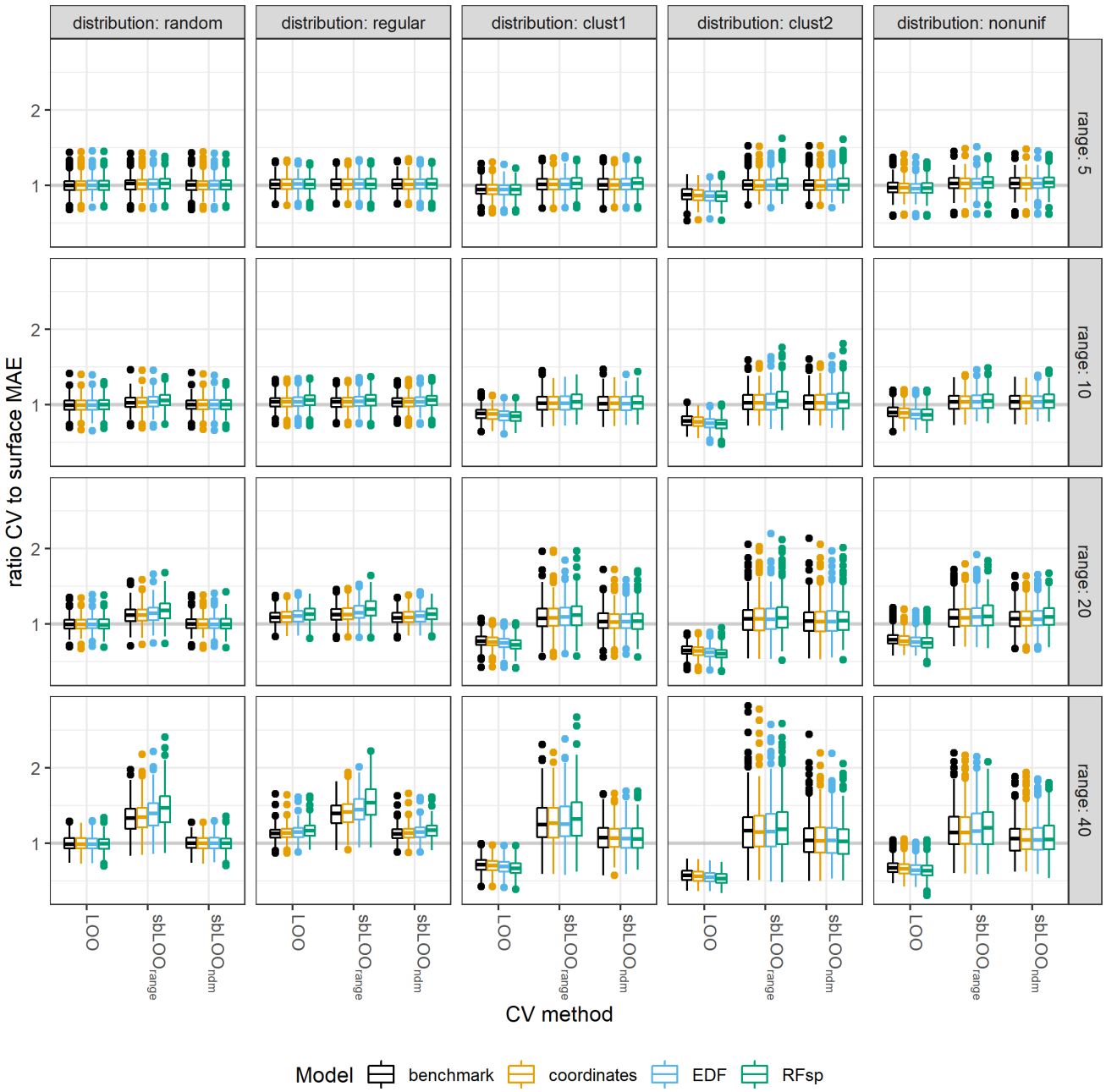


Figure B.6: Ratio of CV to surface MAE of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

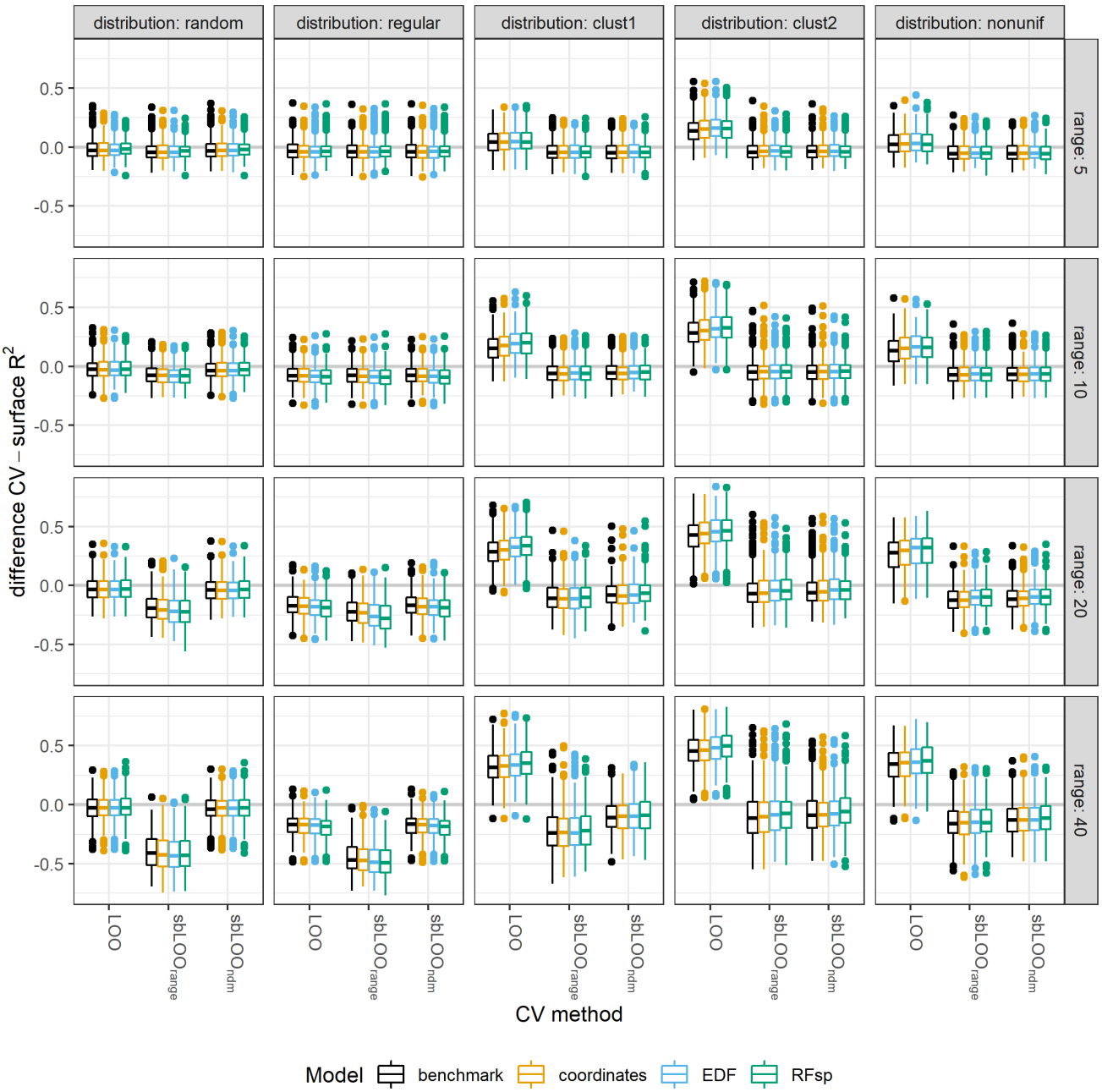


Figure B.7: Difference of CV minus surface R^2 of ML spatial interpolation models by sampling distribution and range, for sample size $n = 100$. Each boxplot consists of 100 data points resulting from 100 sandbox iterations.

List of Figures